PAPERS
IN COMPUTATIONAL LEXICOGRAPHY
COMPLEX '94

Edited by
Ferenc Kiefer, Gábor Kiss and Júlia Pajzs

RESEARCH INSTITUTE FOR LINGUISTICS
HUNGARIAN ACADEMY OF SCIENCES, BUDAPEST

PAPERS IN COMPUTATIONAL LEXICOGRAPHY
COMPLEX '94

PAPERS IN COMPUTATIONAL LEXICOGRAPHY COMPLEX '94

Edited by
Ferenc Kiefer, Gábor Kiss and Júlia Pajzs

RESEARCH INSTITUTE FOR LINGUISTICS
HUNGARIAN ACADEMY OF SCIENCES, BUDAPEST
1994

Proceedings of the 3rd International Conference on
Computational Lexicography, COMPLEX '94
Budapest, Hungary

All correspondence should be sent to
Research Institute for Linguistics
Hungarian Academy of Sciences
Department of Lexicography and Lexicology
Budapest P.O. Box 19
Hungary 1250

Cover design by Gábor Kiss
Technical assistence: Judit Pais

ISBN 963 8461 78 0

Hozott anyagról sokszorosítva

9421541 **AKAPRINT** Nyomdaipari Kft. Budapest. F. v.: dr. Héczey Lászlóné

Contents

PREFACE

This volume collects the papers presented at the third conference on Computational Lexicography and Text Research, held at Budapest, on 7–9 July 1994. The conference was jointly organized by the Hungarian Academy of Sciences, Research Institute for Linguistics and the Université Paris 7, Laboratoire Automatique Documentaire et Linguistique. This time again a great number of papers were submitted for the conference ranging through most topics of computational lexicography and corpus research: from the very theorethical subjects of lexicography such as prototype theory to such practical problems as, for instance, optimal methods for parallel search in the lexicon. Some papers are on different experiences in corpus research, a couple of them offering a method for lexical disambiguation, sense discrimination. We can see interesting examples of using and building lexical databases for different purposes (MT, AI, speech generation and recognition etc.). The new possibilities offered by electronic publishing of dictionaries are also presented.

We are grateful for the work of the program committee who assited in selecting among the submitted papers and helped the authors to prepare the final version of their presentation by their valuable comments. The members of the committee: *Anna BRAASCH* University of Copenhagen, *Maurice GROSS* Université Paris 7, *Ferenc KIEFER* Hungarian Academy of Sciences, *Ole NORLING-CHRISTENSEN* University of Copenhagen, *Júlia PAJZS* Hungarian Academy of Sciences, *Tamás VÁRADI* University of London.

Júlia Pajzs

A User-Centered Meta-Formalism for Morphology

STEPHAN BOPP – MARC DOMENIG

Abstract

This paper presents a system for the specification and the use of dictionary databases. Prominent characteristics of the system are its user-centredness and its knowledge specification formalism. We call the latter a meta-formalism because it allows the linguist to work on a higher level of abstraction than formalisms based on rewriting rules. The paper focuses on the "tool-character" of the system rather than on the underlying algorithms. The formalism has been implemented and tested in several prototyping cycles. Specifications of Italian, German, English and French morphological rule bases have shown that the system is particularly promising for the formalization of word formation.

1. Introduction

The system Word Manager can be viewed as a successor of Koskenniemi's two-level model (Koskenniemi 83). Unlike most systems inspired by the two-level model (Bear 88, Emele 88, Görz 88, Kataja 88, Kay 87, Koskenniemi 90, Trost 90, Karttunen 92), Word Manager does not try to extend the formalism's expressiveness for a wider coverage of languages, but it was originally designed to improve the data management capabilities of the system (Domenig 89, 90). The implementation of a first fully operational prototype version in 1989 and three major redesigns resulted in a system with, c.a., the following characteristics:

- Word Manager follows a client-server model, where a server handles the data management and different clients handle the data access - including all user interfacing.
- Focus on reusability: Word Manager maintains a large network of knowledge accessible in various ways (see below). The purpose of this approach is to construct a reusable database: the database must be accessible by all kinds of applications requiring morphological knowledge.
- Distinction between rule and entry knowledge: a sharp distinction is made between the specification of rule and entry knowledge. Rule knowledge has to be specified before entry knowledge can be added. The system distinguishes separate user interfaces for the specification of the two kinds of knowledge: they are called linguist interface and lexicographer interface, respectively.

The following sections will focus on the user-centredness of the system. We will show this with the example of the linguist interface, a version of which is in the public domain and installed on an ftp server.

2. Tools for the Morphological Knowledge Specification

The linguist is responsible for the specification of morphological rules. The linguist interface is the client who has full access to the knowledge specification formalism of Word Manager (WM). The description of the tools available for the specification of morphological rules will demonstrate that the system has been designed to meet the requirements of a linguistic expert in a user-friendly manner.

2.1. Database as Document

A WM-Database is a morphological dictionary database consisting of morphological rules and entries, usually but not necessarily, of one language. Each database corresponds to one document that can be manipulated only from within a dedicated "knowledge engineering environ-

ment". This environment supports the system's formalism by providing dedicated editors for a number of sub-formalisms, each of which covers a specific domain. In addition, it offers testing and debugging tools which permit the user to work in specification/compilation/testing cycles.

2.2. Structuring of the Specification

The user can structure the rules hierarchically into so-called inflection units and word-formation units. She or he can use the same or similar structuring criteria as in a traditional grammar. This both facilitates the specification process and results in specifications that are easy to understand. Figure 1 shows the structuring of a comprehensive Italian morphology (Bopp 1993).

Italian:inflection	Italian:word-formation

root
  (Cat N)
  (ICat Regular)
  (ICat Irregular)
  (ICat Hard-Coded)
  (Cat Adj)
  (Manner Qual)
  (ICat Regular)
  (ICat Irregular)
  (ICat Hard-Coded)
  (ICat Indic)
  (ICat Entered)
  (ICat Hard-Coded)
  (Manner Poss)
  (Cat V)
	root
	  (WFCat Derivation)
	  (WFCat N-To-A)
	  (WFCat A-To-N)
	  (WFCat N-To-N)
	  (WFCat A-To-A)
	  (WFCat N-To-V)
	  (WFCat V-To-N)
	  (WFCat Conversion)
	  (WFCat Suffixing)
	  (WFCat A-To-V)
	  (WFCat V-To-A)
	  (WFCat A-To-Adv)
	  (WFCat NCF+Suffix)
	  (WFCat Compounding)

Fig. 1: Outline structure of inflection units and word-formation units (not fully extended; features in bold have underlying sub-nodes)

2.3. Local Specification Process

The hierarchical structure enables the linguist to work "locally". The inflection units and the word-formation units are largely independent. Furthermore, it is possible to restrict the scope of rules to one or any number of sub-units, to one particular type of entry, etc. In this manner, highly generalizing rules as well as rules with a clear-cut "local" scope can be specified. Consider the string manipulation rule responsible for the umlauted plural in German nouns like "Vater/Väter" ('father/fathers'):

```
(ISRule Noun-Umlaut_a/ä)
"(.*)A(.*)/\1ä\2" (ICat N-Stem)          (ICat N-Suffix)(Num PL)
```

The example shows that the formalism used for string manipulation rules allows restrictions on the strings (input must contain the character "A") as well as restrictions on the features of the formatives that are combined into a noun plural form (a noun stem plus a noun plural suffix). The rule is fired only when all the restrictions are met ("A" is replaced by "ä"). Further restric-

tions on such string manipulation rules can be defined by associating them with individual inflection rules, word-formation rules, or even with single entries. The latter possibility can be employed for irregular phenomena and as an "escape hatch" for cases where an exception is discovered at a late stage in the data acquisition process.

2.4. The meta-formalism

The Word Manger formalism can be considered a meta-formalism: it abstracts away from the underlying machine-oriented processing in order to provide a user-centred view. Let us illustrate this with two example rules.

The first rule is an inflection rule for Italian nouns of the a/e-class (e.g. "donna/donne" 'woman/women'; "pizza/pizze"):

```
(RIRule   N-Regular.+a/+e)

citation-forms
(ICat N-Stem)  (ICat N-Suffix.+a)(Num SG)

word-forms
(ICat N-Stem)  (ICat N-Suffix.+a)(Num SG)
(ICat N-Stem)  (ICat N-Suffix.+e)(Num PL)
```

The rule combines a noun stem and a singular suffix to the wordform of the singular and the same stem plus a plural suffix to the wordform of the plural. The formatives (stem and suffixes) are specified within the same inflection unit (cf. 2.2.) as the rule. The features specifying formatives and rules can be chosen freely, the only restriction being that regular inflection rules have the attribute 'RIRule', regular word-formation rules the attribute 'RWFRule', etc. This, again, allows the user to keep the specification very close to the terminology used in traditional linguistics.

The second example rule is a word-formation rule defining the prefixing of regular Italian nouns (e.g. "presidente > vicepresidente", "formalismo > metaformalismo"):

```
(RWFRule   Derivation.To-N.N-To-N.Prefixing)
source
1       (WFCat Prefix)
(Cat N) (RIRule ?) >          entry-features (Gender >)
        2       (ICat N-Stem)
target
(RIRule ?)
        1 2     (ICat N-Stem)
```

All noun stems belonging to a lexeme class defined by an inflection rule for regular nouns (Cat N)(RIRule ?) can be prefixed with formatives qualified by a feature (WFCat Prefix)

(specified within the same word-formation unit). The prefix (1) and the noun stem (2) are combined into a noun stem in the order defined by the digits representing them under *target*. The lexeme classes of the newly formed entries and their gender features are propagated (indicated by ">") from the source lexemes.

The examples show why linguists understand this formalism after a relatively short learning period: they can use familiar terminology and knowledge factoring. Furthermore, the examples illustrate why we call the formalism a meta-formalism: the rules are on a higher level of abstraction than rewriting rules. This means that they can be compiled in various ways. Currently, WM supports three types of compilation: First, it generates a network which can be used by a finite-state machine for the analysis and generation of inflected forms. Second, it compiles a set of AI-type condition/action rules which permit the generation of word formations. These rules are primarily employed to enter complex entries, so that derivational dependencies between lexemes can be recorded and controlled by the system. The third compilation algorithm generates a set of rewriting rules that can be interpreted by a unification-based parser, which permits analysing complex words that are potentially generated by the word-formation rules. Contrary to most hand-compiled rewriting rules for word-formation analysis, these rules are not designed to construct parse trees containing morphosyntactic information; instead, they build trees whose nodes represent meta-level word-formation rules, which means that they can be used for (semi-)automatic registration of complex entries (see 2.7).

Evidently, further rule sets could be derived from WM's meta-level formalism. Given our current rule generation algorithms, it would be quite easy, for instance, to compile a set of rules that collects morphosyntactic features for unknown word formations.

2.5. Browsing Facilities

To support the user in the task of specifying knowledge, sophisticated browsing options were realised. They offer the possibility to view and access the specified knowledge from different perspectives.

2.5.1. General Entity Browser

The General Entity Browser allows the user to browse the entire network of entities: rules, formatives and entries. By indicating both the kind of entity (Entity Restriction) and a restriction on the features qualifying these entities (Feature Restriction), the user can search very selectively. Figure 2 shows the result of browsing with a restriction on the entity RlRules (Regular Inflection Rules) and the feature (Cat N) (Category Noun) in a German database:

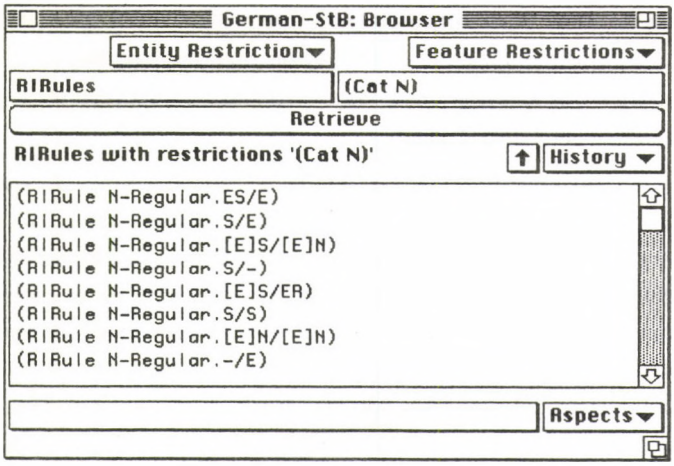


Fig. 2: Entity Browser with RIRule selection for nouns

Each of these entities can be further explored with the Aspects menu in the lower right-hand corner of the browser. The user clicks on one of the listed entities and selects one of the options in the aspects menu. The aspects available are different, of course, for different kinds of entities. Figure 3 shows (some of) the entries inflecting with the rule (RIRule N-Regular.+ES/+E):

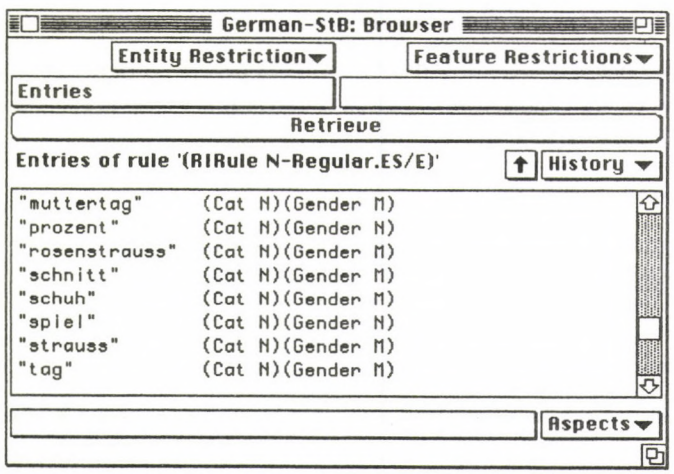


Fig. 3: Entries inflecting with the inflection rule (RIRule N-Regular.+ES/+E)

Since the retrieval of further entities results in their being collected and displayed in the table the "original" entity was displayed, this retrieval procedure can be repeated infinitely.

2.5.2. Lexeme Browser

This browser is specially designed for testing purposes. It can be invoked from the General Entity Browser or by analysing words. It offers different views on a lexeme: the user can test its inflection rule (e.g. by viewing the wordforms it generates, cf. Fig. 4), the word-formation rule by which the lexeme has been created, the word-formation rules by which other lexemes have been derived or composed (e.g. by viewing the so-called Generation History, cf. Fig. 5), etc.

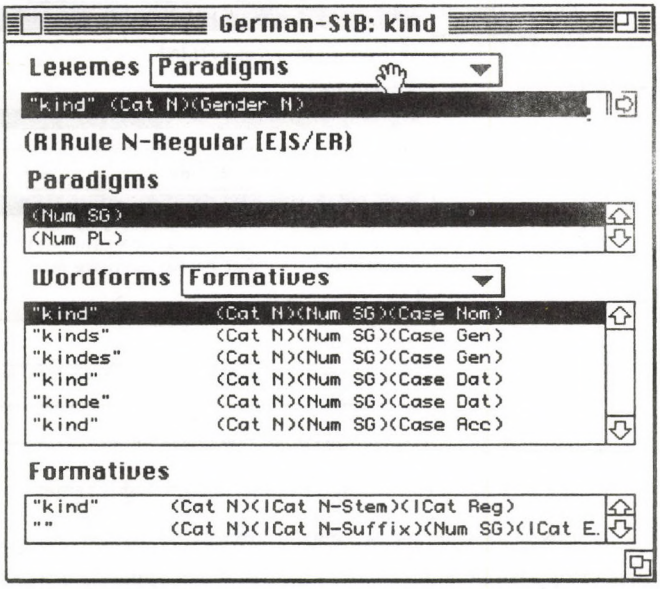


Fig. 4.: Lexeme Browser of "kind" ('child'), view on wordforms of the singular

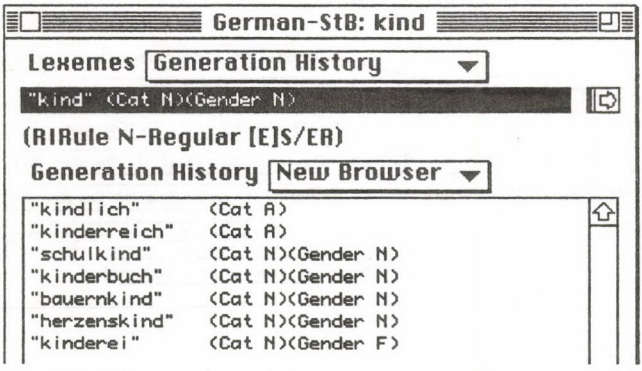


Fig. 5: Lexeme Browser of "kind", view on derived lexemes

By selecting one of the words listed under Generation History, a further lexeme browser can be opened. In this way, the user can follow step by step all dependencies between lexemes (fig. 6)

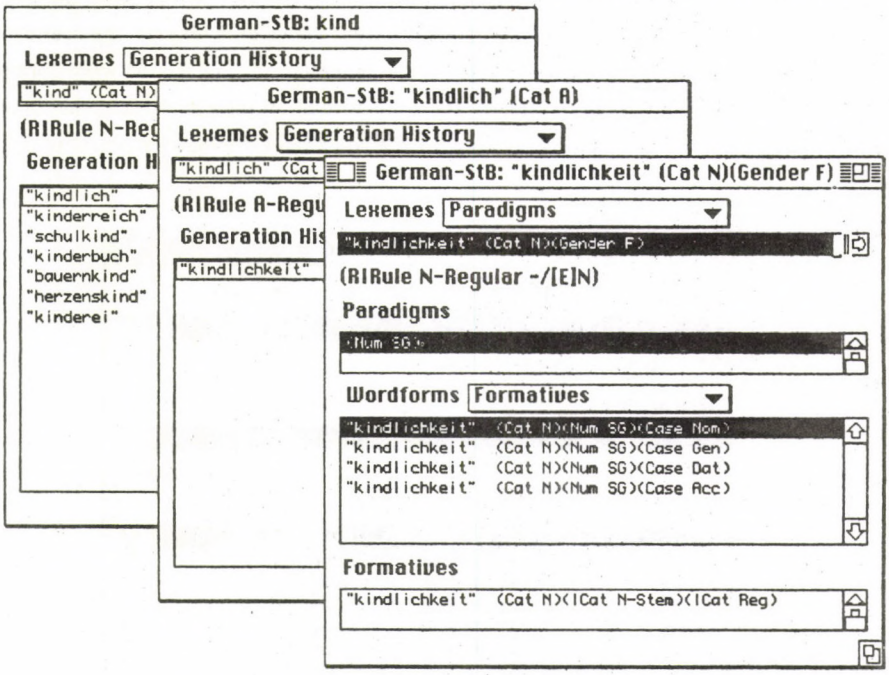


Fig. 6: Sequence of lexeme browsers of related lexemes

Further viewing options show whether two entries are related by derivation, compounding or conversion, what string manipulation rules were fired when applying a word-formation rule, the dependency relations between entries (Fig. 7), etc.

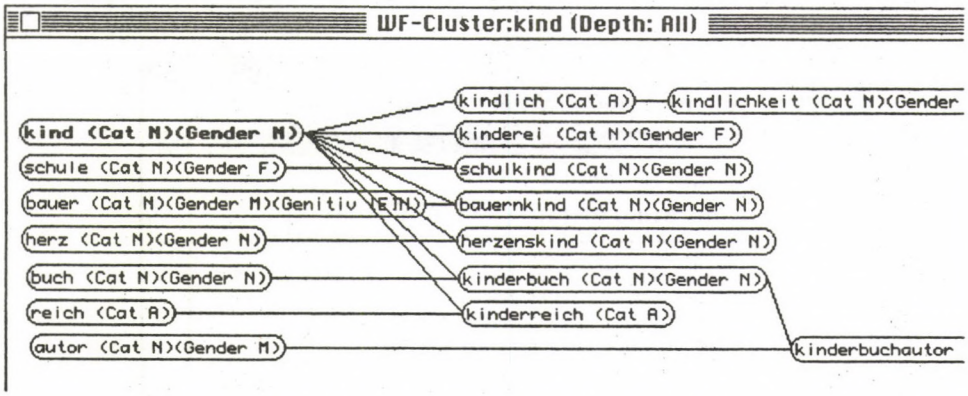


Fig.7: "kind": (partial) word-formation cluster view

2.6. Test Entries

As the examples above illustrate, a WM-database contains entries in the rule specification phase already. Each specification of rule knowledge includes a number of so-called hard-coded entries. They serve two purposes: 1) as test and example entries for particular lexeme classes and 2) for the hard-coding of entries considered irregular. For additional testing, the linguist can temporarily add so-called Lexicographer Entries (LE). By adding simplex entries, the inflection rules are tested; by adding complex entries, the word-formation rules are tested. Since LE are not stored as a part of the rule specification, the user can specify as many LE as he/she wishes without unnecessarily blowing up the rule specification.

2.7. Analysis of Potential Entries

A further test function is provided by the possibility to analyse potential entries. These are entries that are not (yet) contained in the database but composed of elements (stems, affixes) which are already stored. The system proposes derivations according to the word-formation rules specified in the database. In the linguist interface, this option can be used to test the completeness of the word-formation rules. Figure 8 shows the derivations proposed for the word-forms "legalized", "uncommonly" and "machine-readable" in an English database.

Correct parses can be selected and directly entered into the database as lexicographer entries (cf. 2.6.)

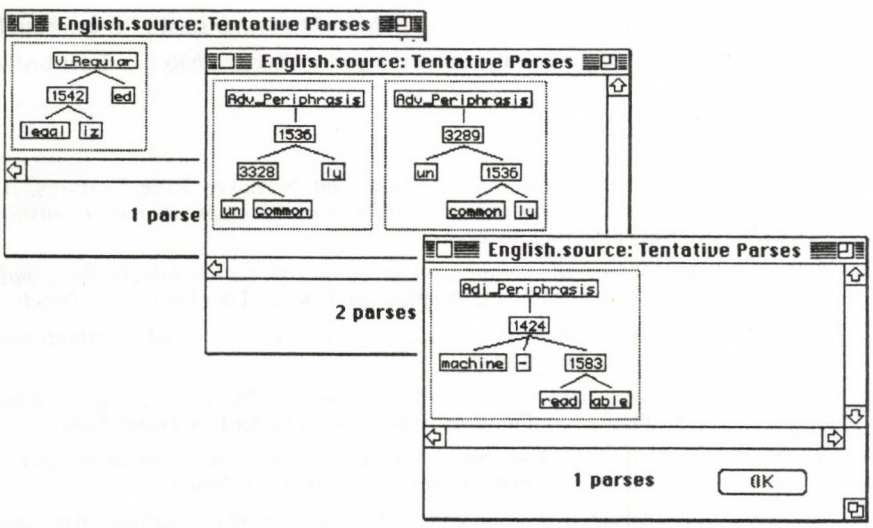


Fig. 8: Tentative parse trees for potential entries

3. Conclusion

We have presented a system we consider a successor of the two-level model. Its design was originally focused on data-management capabilities, which resulted in a client-server architecture and a formalism with two distinctive characteristics: user-centredness and the introduction of a meta-level which abstracts away from rewriting rules. While the first characteristic provides obvious benefits for the end user, the latter is promising because it carries the potential of compiling the meta-level rules to different types of data structures and rules for different purposes. So far, three compilation algorithms have been realized, all of which serve primarily for rule and entry acquisition purposes. Other algorithms, optimised for run-time usage of operational databases, have yet to be conceived.

Several comprehensive morphological rule bases have been developed. This experience has shown that the system is both easily understandable for linguists and powerful enough to allow the specification of the inflectional and derivational morphology of several natural languages (Bopp 88, 93), (Brunner 91), (Garcia 91), (Gregorio 93), (Gupta 89). The rule base for Italian morphology (Bopp 93) is - specially as far as word formation is concerned - the most comprehensive of this language.

The database structure, the sophisticated specification facilities and the flexible knowledge representation resulted in a large system that was expensive to develop. But then, Word Manager is conceived as a system to be used in a larger client-server environment. Furthermore, it is possible to transpose the knowledge contained in a WM-database into small, PC-compatible systems like, e.g., the morphological analysers developed at Xerox PARC as described in Karttunen (1992).

References:

Bear J. (1988): 'Morphology with Two-Level Rules and Negative Rule Features.' In *Proceedings of the 12th International Conference on Computational Linguistics, COLING-88,* Budapest, August 22-27.

Bopp S. (1988): *Tentativo di formalizzazione computazionale della morfologia flessionale dell'italiano*, Lizentiatsarbeit an der Philosophischen Fakultät I der Universität Zürich.

Bopp S. (1993): *Computerimplementation der italienischen Flexions- und Wortbildungsmorphologie*, Olms Verlag, Hildesheim.

Brunner C. (1991): *An Implementation of English Morphology Using the Program Word Manager*, Lizentiatsarbeit an der Philosophischen Fakultät I der Universität Zürich.

Domenig M. (1989): *Word Manager, A System for the Specification, Use and Maintenance of Morphological Knowledge*, Habilitationsschrift, University of Zurich.

Domenig M. (1990): 'Lexeme-based Morphology: A Computationally Expensive Approach Intended for a Server-architecture', in *Proceedings of the 13th International Conference on Computational Linguistics COLING-90,* Helsinki.

Domenig M., ten Hacken P. (1992): *Word Manager: A System for Morphological Dictionaries*, Olms Verlag, Hildesheim.

Emele M. (1988): 'Überlegungen zu einer Two-level Morphologie für das Deutsche.' In *Proceedings 4. Oesterreichische Artificial-Intelligence-Tagung*, Wien, August 29-31, 1988. Published in the series *Informatik-Fachberichte*, 176, Springer.

Garcia C. (1991): *Computerimplementation der deutschen Morphologie*, Lizentiatsarbeit an der Philosophischen Fakultät I der Universität Zürich.

Görz G., Paulus D. (1988): 'A Finite State Approach to German Verb Morphology.' In *Proceedings of the 12th International Conference on Computational Linguistics, COLING-88*, Budapest, August 22-27.

Gregorio S. (1993): *Implementation of English Inflectional and Derivational Morphology*, Lizentiatsarbeit am Institut für Informatik der Universität Basel.

Gupta A. (1989): *La formalisation de la morphologie française sur la base du système Word Manager*, Lizentiatsarbeit an der Philosophischen Fakultät I der Universität Zürich.

Karttunen L., Kaplan R. M., Zaenen A. (1992): 'Two-Level Morphology with Composition.' In *Proceedings of the 15th International Conference on Computational Linguistics, COLING-92*, Nantes, July 23-28.

Kataja L., Koskenniemi K. (1988): 'Finite-state Description of Semitic Morphology: A Case Study of Ancient Akkadian.' In *Proceedings of the 12th International Conference on Computational Linguistics, COLING-88*, Budapest, August 22-27.

Kay M. (1987): 'Nonconcatenative Finite-State Morphology.' In *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics*, Copenhagen, April 1-3.

Koskenniemi K. (1983): *Two-Level Morphology: A General Computational Model for Word-Form Recognition an Production*, doctoral thesis at the University of Helsinki, Publications N° 11.

Koskenniemi K. (1990): 'Finite-state Parsing and Disambiguation.' In *Proceedings of the 13th International Conference on Computational Linguistics, COLING-90*, Helsinki, August 20-25.

Trost H. (1990): 'The application of two-level morphology to non-concatenative German morphology.' In *Proceedings of the 13th International Conference on Computational Linguistics, COLING-90*, Helsinki, August 20-25.

The Organization of the Lexicon in GSF: Structure and Implementation

LORNE H. BOUCHARD - LOUISETTE EMIRKANIAN

A wide-coverage computational grammar must be based on a comprehensive lexical database in which the information is structured and represented in an efficient way. We describe how morphological, syntactic and semantic knowledge of French can be extracted systematically and in a computationally economical way from standard reference works such as *Le Grand Robert de la langue française* and *Le dictionnaire de notre temps* in order to construct a lexical database. This task is a bootstrap process, whereby certain information which can be gleaned easily from the dictionary is used to glean further information, and so on through multiple stages. The ultimate goal is to construct a lexical knowledge base which can account for language regularity in a systematic way and can cope with the creative use of language.

INTRODUCTION

This research was undertaken as part of a larger project, the goal of which is to develop a wide-coverage computational grammar of French [Emirkanian & Bouchard 1992]. From a practical point of view, a computational grammar must have wide-coverage if it is to be truly useful in the large. From a more theoretical point of view, a computational grammar must also have wide-coverage if it purports to be a credible model of natural language performance [Bouchard, Emirkanian & Morin 1992]. The importance of the lexicon as a central repository of phonological, morphological, syntactic and semantic information is stressed in most contemporary linguistic theories. Hence a wide-coverage computational grammar must be based on a comprehensive lexical database in which the information is structured and represented in an efficient way. The construction of such a database is a considerable task and must be

automated as much as possible, or computer assisted at the very least. Furthermore, this task is open ended since the system must constantly learn new words and be able to augment and refine existing entries. Ultimately such a system should be capable of accounting for novel or creative use of language, a formidable task indeed.

SUBCATEGORIZATION IN GPSG AND GSF

Lexical ID rules are the components in Generalized Phrase Structure Grammar (GPSG) [Gazdar, Klein, Pullum & Sag 1985] which bridge the gap between grammar and lexicon: subcategorization information is a codification of lexical behavior. Subcategorization of the head of lexical ID rules is used to encode the complement structure not only of verbs, but also of adjectives and nouns. Because of space limitations, we shall focus exclusively on verbs for the remainder of our presentation. The complement structure is a complex reflection in syntax of distinctions based upon simpler semantic properties of verbs [Levin 1993].

Verb subcategorization has been treated extensively in the Grammaire Syntagmatique du Français (GSF) [GIREIL 1993]. Subcategorization frames or schema are associated with SUB features and associated with a given verb is a list or set of such features. Figure 1 shows a partial list of the subcategorization codes representative of those used in the GSF.

Features	Corresponding schema	Examples
SUB0	Intransitif	Dormir, travailler
SUB1	N3	Couper, manger, résoudre, apercevoir, tenir
SUB2	N3,(N1)	Nommer, élire
SUB3	N3,(P3[à,CPR N3])	Fournir, apporter, envoyer, dire
SUB4	N3,(P3[de,CPR N3])	Retirer, extraire
SUB5	N3, ADV3[avec, CPR N1]	Traiter Marie avec soin
...		
SUB20	P3[à,CPR N3]	Mentir, penser, attentif, profiter,accès, tenir
...		
SUB33	P3[à,CPR V3]	Donner, penser, songer, tenir, contribuer
SUB34	P3[de,CPR V3]	Jurer, douter, profiter, arrêter
...		
SUB50	Q3[que]	Aimer, vouloir, fait, penser
SUB51	V3[VFORM er]	Aimer, vouloir, apercevoir, penser, pouvoir

Representative list of subcategorization schema used in GSF

Figure 1

The verb *penser* , for example, has the features SUB20+, SUB33+, SUB50+ and SUB51+. Figure 2 lists the subcategorization frames associated with *penser*.

Features	Corresponding schema	Examples
SUB20	P3[à,CPR N3]	Gilles pense à Mireille
SUB33	P3[à,CPR V3]	Gilles pense à visiter Boston
SUB50	Q3[que]	Mireille pense que cette thèse est très bonne
SUB51	V3[VFORM er]	Mireille pense venir

Subcategorization schema for *penser*

Figure 2

This information was painstakingly compiled by hand for the lexicon used in the prototype of the GSF and it was felt that somehow this task must be automated or computer-assisted at least when scaling up the prototype. Furthermore, although GSF is currently a morphosyntactic analyzer, we plan to extend it with a semantic component, since all problems in automatic language analysis cannot be solved by morphosyntax alone [Bouchard & Emirkanian 1992]. This involves storing more information in the lexicon. In particular, sortal restrictions [Alshawi & Carter 1992] which are part of the knowledge of language which lies on the boundary between syntax and semantics. Sortal restriction information is intimately linked with case marking information and can be considered a refinement thereof. Beyond sortal restrictions, simplified semantic analysis requires the thematic structure of verbs.

To our knowledge, there are currently no comprehensive machine-readable lexicons for the French language which are generally available and it is somewhat reluctantly that we decided to construct a lexical database.

Machine-readable dictionaries are sources of natural language knowledge in which information is stored in a coherent and systematic way. The knowledge extracted can be used to build a lexical database, a necessary first step towards building a lexical knowledge base. Also, although most existing dictionaries were constructed for consultation by human readers, we think it is interesting to explore how readable they are by a machine, the purpose of which is to extract specific knowledge of language in a systematic and efficient way.

STRUCTURE OF THE LEXICON FOR GSF

The organizing principle of lexical knowledge in GSF is that of a lattice with multiple inheritance and default values, principles which have been widely adopted by the knowledge representation community in artificial intelligence [Brachman, Fikes & Levesque 1983]. Inheritance is a principle which allows common information to be stored once, at the highest level possible in the hierarchy, and to be shared by all items which inherit it, unless this is explicitly

overridden locally. Pure simple inheritance can be shown to implement, in an efficient manner, a simple form of logical inference. With multiple inheritance an item can inherit from more than one ancestor. Although multiple inheritance can be shown to be problematic, especially in non-monotonic contexts, it can be used to structure the lexicon by effectively eliminating redundancy provided it used in a disciplined way [Russell, Ballim, Carroll & Warwick-Armstrong 1992]. Finally, default values are a convenient technique for specifying a value which is to be used as a default value, that is unless it is explicitly stipulated to be otherwise. The use of default values can also help reduce redundancy. We seek to construct this lattice structure with the help of the computer, which means that the relevant data must be available in a structured machine-readable form. This data is extracted mainly from machine-readable dictionaries.

KNOWLEDGE EXTRACTION: A BOOTSTRAP PROCESS

The knowledge we seek to extract from machine-readable dictionaries is essentially of three types, i.e., morphological, syntactical and semantic. This extraction is implemented in steps. The extraction process is assisted by a number of tools which we have implemented.

Browsing the dictionary

Although the analysis of *Le dictionnaire de notre temps* had been performed in a UNIX environment, we chose to analyze *Le Grand Robert* on the Macintosh using HyperCard 2.2, which supports the international character sets, since we had a wealth of existing scripts and pre-compiled external commands (XCMDs) at our disposal. This HyperCard-based system now forms the core of our workbench for exploring French lexical data.

Le Grand Robert on CD-ROM is split into two main files: a definition file (47 Mb) and a citations files containing quotes from French literature (32 Mb). The dictionary file is pre-indexed by the list of word entries (nomenclature). However both files can be searched on-line in our system as free text and the fact that they can both be indexed on-the-fly, either in full or partial word mode, turns out to be invaluable in practice. Indeed, we were so pleased with the results that we also have created a HyperCard stack for *Le dictionnaire de notre temps*. A screen dump of the entry for *abaisser* can be found in Figure 3 on the next page.

Word tagging

The nomenclature provides the grammatical function of the words directly, however it is incomplete in the sense that even some frequently occurring words are missing. We use a list of the most-frequently occurring words [Catach 1984] which is consulted before the dictionary. There remains of course the problem of dealing with the inflected forms of a word.

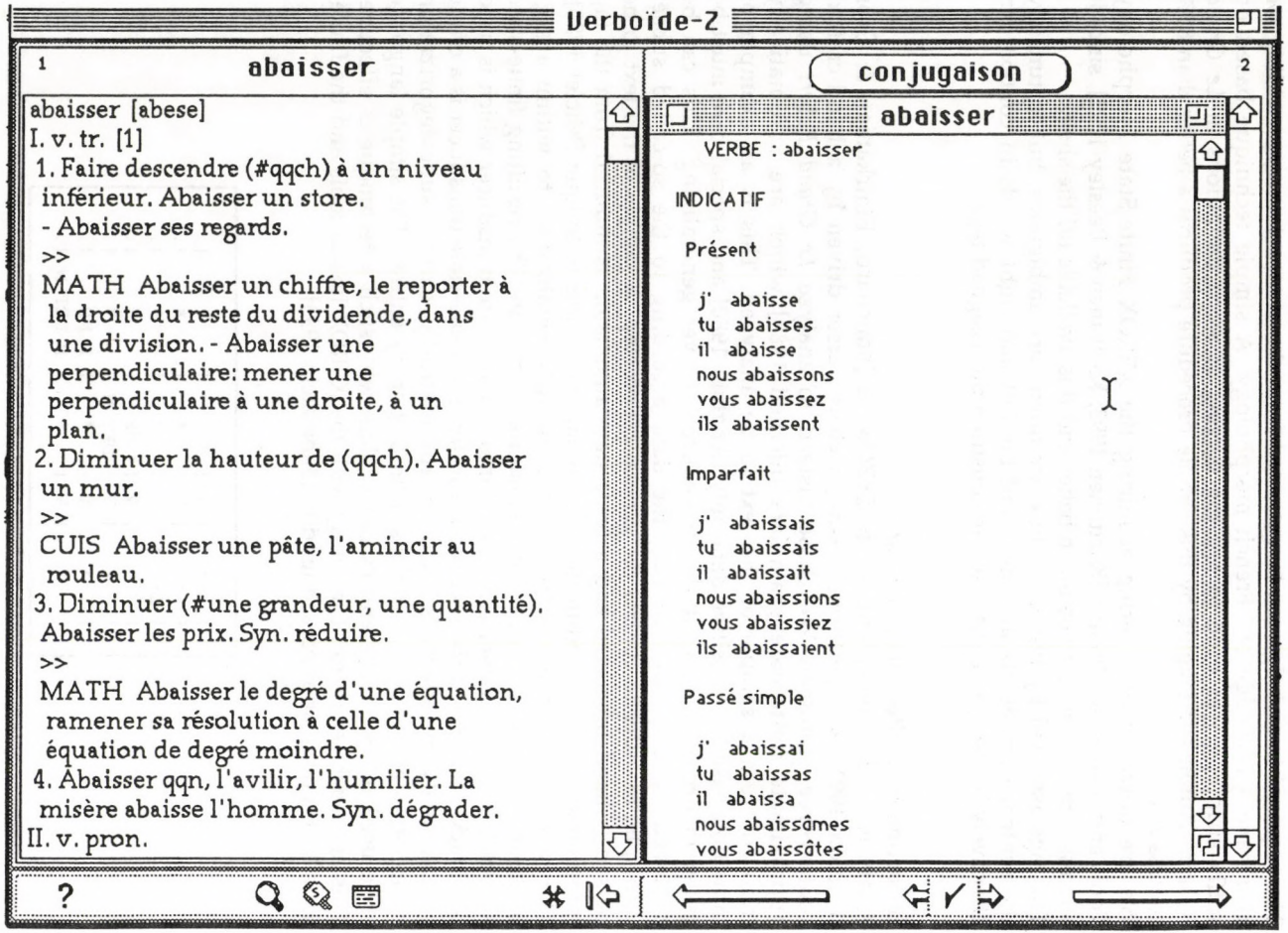


Figure 3 The entry for *abaisser*

The required knowledge can be extracted from the nomenclature, providing we also have knowledge of French morphology. A simple technique based on suffix stripping was used to test the tagging of word definitions in *Le Grand Robert*. Automatic tagging by this simple procedure produces a better than 85% success rate.
We are however considering acquiring the XEROX Finite-State Morphology Tools and lexicon for French [Karttunen 1993; Karttunen & Beesley 1992], since it success rate is claimed to be much better and it is available off the shelf.
The tags assigned by this procedure are often very ambiguous, but fortunately the sentence fragments are short and the left and right words in context can effectively be used, in most cases, to constrain the assigned tags.

Induction of finite-state grammars
Analysis of dictionary entries in ZYZOMYS [Bouchard, Emirkanian & Gros d'Aillon 1991] was performed using a chart parser driven by a hand-crafted context-free grammar. We chose instead to analyze *Le Grand Robert* using approximate finite-state grammars [Ejerhed 1988] which are automatically induced from a sample of the text to be analyzed. This is an example of induction based on positive data only [Angluin 1980] and special care must be taken in order to prevent the procedure for over generalizing. This can be achieved by carefully ordering the data according to the so-called subset principle, in order to control the generalization step. A sample of the text to be parsed is hand bracketed and a finite-state automaton is induced from the tag information using a modification of the tail clustering technique [Miclet 1980]. The modification consists simply in limiting generalization to within a segment: this greatly reduces the combinatorial explosion. The resulting finite-state automaton is then converted by hand into a finite-state transducer which is used to bracket the rest of the text automatically. The finite-state transducer is a crude syntax analyzer which is used to automatically extract subcategorization information and sortal restrictions from dictionary entries. The simple language style used in dictionary entries can explain why such a technique is effective. This approach seems to fit in nicely with the XEROX lexical tools and the finite-state local grammar approach used in [Silberztein 1993].

```
|   *    | V |   *    |    *     |.
|  NP    | V |   *    |    *     |.
| DET N  | V |   *    |    *     |.
|   *    | V |  NP    |    *     |.
|   *    | V | DET N  |    *     |.
|   *    | V |   *    | P NP     |.
|   *    | V |   *    | P DET N  |.
```

A sample training sequence for grammar induction

Figure 4

KNOWLEDGE EXTRACTED

Subcategorization information
The transitive/intransitive distinction is systematically recorded in the nomenclature of *Le Grand Robert*. However, there appears to be no systematic way in which the dictionary records case marking information: indeed, sometimes the information is found as part of the word sense definition, sometimes as part of the example. Apart from this ambiguity, case markings which are recorded are easily extracted from either of these sources.

Sortal restrictions
The construction of the sort hierarchy is another example of a bootstrap process. A shallow lattice of sorts patterned on [Alshawi, et al. 1992] is used as a first approximation. The analysis of word sense definitions and of the examples is used to extract the nominal head of the noun phrases which are arguments of a verb and these nouns are then used to refine the lattice. This step currently requires user assistance since *Le Grand Robert* is a language dictionary and lacks the world knowledge coverage of an encyclopedic dictionary. We are considering using *ZYZOMYS* to help reduce the amount of user intervention required.

A DATABASE OF MOTION VERBS

We decided to construct a lexical database of verbs which could be analyzed systematically with the help of the computer. This lexical database is based on information extracted from the machine-readable dictionaries as well as information extracted from the tables of the lexique-grammaire project. The lexique-grammaire project of the LADL of the University Paris VII [Boons, Guillet & Leclère 1976; Gross 1975; Guillet & Leclère 1992; Leclère 1989] has over the years produced a systematic classification of French verbs in the form of tables of their syntactic and semantic properties [Leclère 1990]. Also available is a list of simple sentences which exemplify the classification scheme used in the lexique-grammaire [Guillet 1990].

We are currently analyzing a subset of verb entries, those of verbs involving motion, in order to study and eventually exploit in a systematic way the syntactic and semantic regularities and interrelationship between verbs in this restricted domain.

The actual definition of this class is rather interesting, since motion verb entries are not marked as such in *Le Grand Robert*. From an initial list of known motion verbs — which incidentally was found in *Le Grand Robert* in the entry for *mouvement* — the list is extended by analyzing definitions in the dictionary. Although word production by affixation is not as regular in French as it can be in other languages, a number of motion verbs can be identified directly in the nomenclature using affix analysis, as for example from the root verb *porter* the following list of verbs are produced by prefixation: *apporter, colporter, déporter,*

exporter, héliporter, etc. The list of synonyms, which can readily be extracted from *Le Grand Robert* and *Le dictionnaire de notre temps* is used to enrich the initial list. The properties of the motions verbs are organized along many dimensions and we are investigating how it can be represented as a lattice with multiple inheritance.
Each entry in the lexical database which describes a verb has a number of fields which include the classes assigned to it in the work of the LADL, subcategorization information, thematic role assignment, focus, list of prepositions, lists of synonyms and antonyms and finally examples sentences of uses of the verbs. This database has approximately 550 entries at the current time. Figure 5 shows a typical entry in this database.

```
VERBE= abaisser
  CLASSES=     {38L}
  SOUSCAT=     <SN:x,SN:y,(SP:z),(SP:w)>
  ROLES=             [x=agent,y=thème,z=source,w=lieu]
  FOCUS=             final
  PRÉP=        INIT=       {de}
               MED=        {}
               FINAL=      {à}
SYN= {baisser, biller, abattre}
ANT= {élever, relever, soulever, surélever, exhausser, hausser}
EX=
  {Voulez-vous abaisser la vitre ?
   Abaisser qqch. en inclinant, en penchant.
   Abaisser un bras, la tête.
   Abaisser la pâte, de la pâte avec un rouleau à pâtisserie,
         l'aplatir en couche mince.}
```

An entry in the motion verb database

Figure 5

CONCLUSION

The design and implementation of a wide-coverage lexicon is a considerable task which requires the availability of many resources: a comprehensive lexical database, computer tools and manpower. We hope that this investment can in some small way contribute to the advancement of more effective automatic natural language processing.

Acknowledgment
We wish to thank all the research assistants involved in the GSF project and in particular André CLOUTIER, Simon PLOUFFE, Benoît ROBICHAUD and Caroline VIEL who were more closely involved with the lexical aspects of the research.

REFERENCES

Alshawi, H. & D. Carter. (1992). Sortal Restrictions. In H. Alshawi (Ed.), *The Core Language Engine* (pp. 173-185). Cambridge MA: The MIT Press.

Angluin, D. (1980). Inductive Inference of Formal Languages from Positive Data. *Information and Control*, **45**, pp. 117-135.

Boons, J.-P., A. Guillet & C. Leclère. (1976). *La structure des phrases simples en français. I: Phrases intransitives*. Genève: Droz.

Bouchard, L., L. Emirkanian & J.-Y. Morin. (1992). Computational Grammar as Knowledge Representation. In *PROC. Sixth International Conference on Systems Research Informatics and Cybernetics (Volume II)*, Baden-Baden, International Institute for Advanced Studies in System Research and Cybernetics, pp. 121-132.

Bouchard, L. H. & L. Emirkanian. (1992). An Exploratory Environment for the Study of the Syntax and Semantics of Natural Language. *Fourth Symposium on Logic and Language*, Budapest.

Bouchard, L. H., L. Emirkanian & F. Gros d'Aillon. (1991). Extracting French Morphological and Syntactic Information from a Machine-Readable Dictionary. In *Computational Lexicography*, Balatonfüred, Research Institute for Linguistics, Hungarian Academy of Science, pp. 9-24.

Brachman, R. J., R. E. Fikes & H. J. Levesque. (1983). KRYPTON: A Functional Approach to Knowledge Representation. Research Report No. 16, Fairchild Laboratory for Artificial Intelligence Research.

Catach, N. (1984). *Les listes orthographiques de base du français (LOB): les mots les plus fréquents et leurs formes fléchies les plus fréquentes*. Paris: Nathan-Recherche, 156 pp.

Ejerhed, E. I. (1988). Finding Clauses in Unrestricted Text by Finitary and Stochastic Methods. In *Second Conference on Applied Natural Language Processing*, Austin TX, pp. 219-227.

Emirkanian, L. & L. H. Bouchard. (1992). Approche computationnelle aux phénomènes morphologiques et syntaxiques du français. Rapport de recherche, UQAM.

Gazdar, G., E. Klein, G. Pullum & I. Sag. (1985). *Generalized Phrase Structure Grammar*. Cambridge MA: Harvard University Press, 276 pp.

GIREIL. (1993). La sous-catégorisation et la cliticisation. Rapport de recherche, Université du Québec à Montréal.

Gross, M. (1975). *Méthodes en syntaxe*. Paris: Hermann.

Guillet, A. (1990). *Phrases simples illustrant les tables de verbes du lexique-grammaire*.Diskette, personal communication.

Guillet, A. & C. Leclère. (1992). *La structure des phrases simples en français: constructions transitives locatives*. Genève: Droz, 445 pp.

Karttunen, L. (1993). Finite-State Lexicon Compiler. Research Report No. ISTL-NLTT-1993-04-02, XEROX Palo ALto Research Center.

Karttunen, L. & K. R. Beesley. (1992). Two-Level Rule Compiler. Research Report No. ISTL-92-2, XEROX Palo Alto Research Center.

Leclère, C. (1989). Les mots ont-ils une grammaire? *Le Français dans le monde*, (numéro spécial intitulé *...Et la grammaire*), pp. 40-49.

Leclère, C. (1990). Organisation du lexique-grammaire des verbes français. *Langue française*, **87**, pp. 112-122.

Levin, B. (1993). *English Verb Classes and Alternations*. Chicago: Chicago University Press, 348 pp.

Miclet, L. (1980). Regular inference with a tail clustering method. *IEEE Trans. on Systems, Man, Cybernetics*, **10**, pp. 737-743.

Russell, G., A. Ballim, J. Carroll & S. Warwick-Armstrong. (1992). A Practical Approach to Multiple Default Inheritance for Unification-Based Lexicons. *Computational Linguistics*, **18**(3), pp. 311-337.

Silberztein, M. (1993). *Dictionnaires électroniques et analyse automatique de textes: le système INTEX*. Paris: Masson, 233 pp.

A Modular and Flexible Architecture for an Integrated Corpus Query System

OLIVER CHRIST

Abstract

This paper describes the architecture of an integrated and extensible corpus query system developed at the University of Stuttgart and gives examples of some of the modules realized within this architecture. The modules form the core of a corpus workbench.

Within the proposed architecture, information required for the evaluation of queries may be derived from different knowledge sources (the corpus text, databases, on-line thesauri) and by different means: either through direct lookup in a database or by calling external tools which may infer the necessary information at the time of query evaluation. The information available and the method of information access can be stated declaratively and individually for each corpus, leading to a flexible, extensible and modular corpus workbench.

1 Introduction

With the availability of tagged and annotated text corpora, corpora cannot be regarded any more as mere sequences of words. Additionally, more and more linguistic knowledge bases become available and provide additional knowledge about words (MRDs, on-line thesauri like WORDNET [Miller *et al.*, 1993], morphological knowledge bases like the CELEX database [Baayen *et al.*, 1993] ,...). When using and querying corpora, all this knowledge should be usable within a corpus query system in order to enable the lexicographer or linguist to express the linguistic properties of the examined phenomenon as precisely as possible (in order to reduce the amount of data which has to be browsed manually), no matter how the knowledge necessary to evaluate the query is stored or by which means it is derived.

When a corpus is thus regarded as a structured object composed of several different knowledge sources, a problem arises because different knowledge sources require possibly different access methods. Furthermore, for many types of information, it is useful not to store the information physically at all but to compute it at the time of query evaluation. For example, bigram tables for large corpora might grow too big to be held online. Automatically assigned part-of-speech tags, on the other hand, might either be stored in a

database when they are regarded as "stable" or might be computed at the time of query evaluation by a tagging tool.

Additionally, a corpus query system need not necessarily be used only by human users: a parser might consult a corpus annotated with parse trees (treebank) to disambiguate between several syntactic structures by looking up similar, but disambiguated syntactic patterns; a generator might use a semantically annotated corpus to filter lexical preferences.

These different knowledge sources, access strategies and usage situations are best supported by a hierarchical, modularized system architecture where the single modules can be combined in different ways to adapt the system to various usage situations.

We therefore designed and implemented the following architecture: To abstract as much as possible from the different storage properties, the data access was split between a "logical data access layer", which is independent of data access methods and storage properties, and a "physical data access layer", which is the data-oriented interface to the knowledge sources and which is responsible for data access and network-based corpus data interchange. The adaptation of the system to different usage situations is achieved through different interfaces to the logical access layer, but tools may also request data from the physical layer directly. A general-purpose query language, which treats the whole corpus as a structured knowledge source and allows to express queries involving all knowledge sources declared for a specific corpus (no matter how the knowledge is accessed physically), was added to the logical access layer. This architecture is sketched in figure 1.

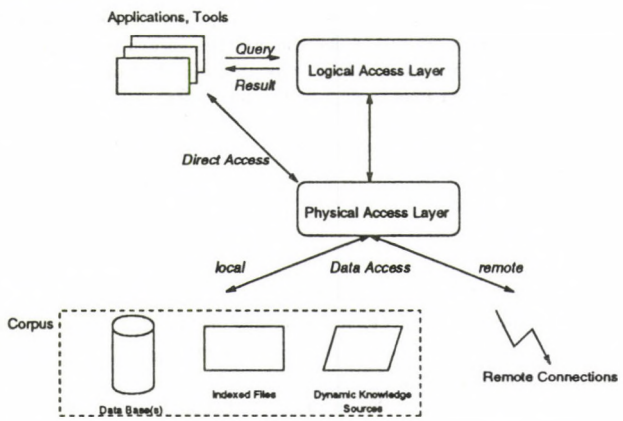


Figure 1: The modular architecture of a flexible corpus query system

In the following sections, these modules are described in more detail. Section 2 outlines the physical layer. In section 3, the logical layer and the query language are described. One usage situation is the interactive use of the query system. For this purpose, presentation and interaction tools have been built which are explained in section 4. In section 5, some directions of our further work are described. The paper ends with a short conclusion in section 6.

2 The physical layer

The task of the physical layer is to provide a uniform interface between the logical layer and the files, databases or tools which "store" the information the corpus is built of. The physical layer therefore encapsulates knowledge about file and tool access and provides an interface which is independent of the storage device and the information type (static vs. dynamic). Due to its proximity to the physical corpus representation, the physical layer also provides methods for corpus management, bigram table creation and management, corpus preparation and indexing, frequency counting etc.

Currently, the physical layer supports the following types of corpus annotations:

- *positional attributes* are attributes where a (string) value is assigned to (almost) every corpus position. The sequence of words the corpus text is built of is one example of a positional attribute. Other examples are part-of-speech tags and base forms (see figure 2). An arbitrary number of positional attributes can be assigned to a corpus;

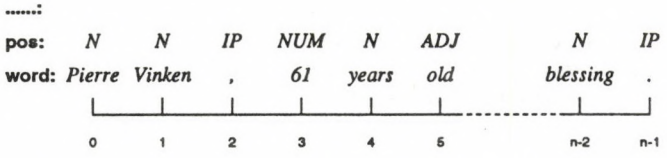


Figure 2: Positional attributes: Values are associated with corpus positions

- *structural attributes* are attributes which capture information about sentence boundaries, article boundaries etc. Currently, recursive structures (like NPs with embedded NPs) cannot be represented. The number of structural attributes is not limited;

- *bigram tables* are related to one of the positional attributes of a corpus and hold information about the absolute number of adjacent occurrences of two values of the attribute within a given window size[1]. Note that, for example, both word bigrams as well as part-of-speech-tag bigrams can be represented;

- *alignment information* can be added to a pair of parallel corpora (which are, roughly speaking, translations of each other) to represent information about corresponding (aligned) ranges (sentences, for example). As in the case of structural attributes, we cannot represent recursive alignments or alignments on more than one level (for example, information about aligned words additionally to aligned sentences);

- finally, *dynamic attributes* are attributes the values of which are not stored physically, but which are computed at query evaluation time by calling external tools, similar to a function call. An arbitrary number of arguments can be declared for a dynamic attribute. When the value of a dynamic attribute is requested, the argument list is filled and an external tool is called. The external tool, then, returns the computed

[1] We use the term *attribute value* to denote one element of the list of distinct strings which occur as the values of a positional attribute. In the case of the corpus text, this is the list of distinct words which occur in the corpus.

value, which is either a string or an integer value. Neiter indices nor bigram tables can be built for dynamic attributes.

A corpus has to be prepared in a special way before its data can be used by the query system. This preparation step involves character set normalization, tokenization, sentence boundary detection (if required), and – in case of annotated corpora – the partitioning of the different positional attributes (for example, corpus text and part-of-speech tags) into several files. Then, a special one-word-per-line format is produced which is used as input for the construction of the internal corpus representation and the indices[2]. The corpus text itself is not needed any more after transforming it into the internal representation. Details of the internal corpus representation and the encoding steps are described in [Christ, 1994].

After a corpus is encoded, it must be *registered*. This is achieved through a *registry file* which declares the attributes and their types assigned to a corpus. All corpus accessing tools access a corpus only via a symbolic name, which is the file name of the registry file. The tools (and the users) therefore need not know where a corpus is stored in the file system in order to access the data. All relevant information is captured in the registry file.

```
NAME "Hansard corpus (english part)"
ID      hansard-e
HOME /corpora/encoded/hansard-e

ATTRIBUTE word
ATTRIBUTE pos

DYNAMIC ishuman(STRING):INT "/corpora/utils/cmd/wn-hypen '$1' human"

ALIGNED hansard-f            # the french part
```

Figure 3: A small sample registry file

A sample registry file may look as illustrated in figure 3. It declares a corpus `hansard-e` and the directory in which the data can be found. Two positional attributes are assigned to this corpus, `word` and `pos`. Additionally, the dynamic attribute `ishuman` is declared, which takes a string as an argument and returns an integer value (where "0" means "no" and "1" means "yes"). Upon query evaluation, a shell command is executed which consults WordNet to evaluate whether the argument string may denote a "human object". The corpus is aligned to another corpus, `hansard-f`.

A corpus can be extended after registration. Positional attributes (as well as all other types of attributes) can be added to an existing corpus without need for reindexing existing data.

For testing purposes, we have implemented a TCP/IP protocol for network-based exchange of corpus data within the physical layer. Through this protocol, it is possible to declare that a given attribute of a corpus (or the whole corpus) is stored on a remote computer. Upon access to remotely stored data, a network connection is built up, access authorization is verified and, if access is granted, the requested data is returned. Through

[2]The internal corpus representation we use is inspired by an – unfortunately – unpublished draft paper by Ken W. Church, "A Set of Unix Tools for Processing Large Text Corpora".

this exchange protocol, it is possible to split corpus data between several computers in the internet. This is useful, for example, to share corpus data between several computers or to run query tools on computers which have too little memory or hard disk space to hold large corpora (although data access is slowed down a lot by remote connections). The remote status of an attribute is hidden within the physical layer, that is, clients of the physical layer do not need to handle remote corpora differently from local data access.

One of the most important "clients" of the physical layer is the logical layer, which is described in the following section. Other clients are tools which do not need to access a corpus through a query language (for example, word list generators or tools which statistically evaluate frequency or bigram counts).

3 The logical layer and the query language

The logical layer uses the information provided by the physical layer to parse and evaluate corpus queries given in the query language described below[3]. Within this layer, the set of positional attributes defined on a corpus can be seen as a sequence of entities referred to by corpus positions. These entities may have several attributes, for example the attribute WORD for the "character string" found at a given corpus position, POS for the part-of-speech tag assigned to that word, ROOT for the base form of that word, etc. The query language allows to find sequences of entities where a number of conditions over such attribute-value pairs hold.

Conditions are boolean expressions which involve attribute-value tests, where all positional attributes defined on a corpus can be used. Such a condition may look as follows:

(1) `[word="chair.*" & pos != "N.*"]`

When this condition is evaluated against a given corpus position, it is tested whether the value of the `word` attribute at that corpus position matches (`=`) the regular expression `"chair.*"` and the value of the `pos` attribute does not match (`!=`) the regular expression `"N.*"`[4].

A query consists of a regular expression over such conditions. In addition to concatenation of conditions, the other standard regular expression operators are available, like "*" for an arbitrary number of repetitions of the preceding regular expression, "+" for at least one repetition, "?" for optionality, and "|" for disjunction. Parentheses can be used for grouping of expressions. [] is a "wildcard" which matches every corpus position. Additionally, the interval operator $\{n, m\}$ is supported, which denotes at least n, but at most m repetitions of the preceding regular expression[5]. Thus, regular expressions are used on the level of attribute values as well as on the level of conditions. Example (1) is already a query, since it is a one-element regular expression.

[3] Currently, the logical layer only supports positional, structural and dynamic attributes; access to bigram and alignment attributes has yet to be implemented.

[4] We use the POSIX EGREP syntax for regular expressions. In this standard, the dot "." matches every character and the star "*" matches any (possibly empty) sequence of the last character or regular (sub-)expression. A common error is to write `"N*"` when all strings beginning with a capital `N` should be matched, but the regular expression `"N*"` denotes all strings which entirely consist of a sequence of capital `N`s.

[5] When m is omitted in such an interval, exactly n repetitions are matched.

When a query is evaluated, the query interpreter computes all matches of the regular expression in the corpus. A match of a query is a "substring" of the corpus, that is, a corpus interval the boundaries of which are the beginning and ending corpus positions of the match. Since regular expressions are used which, in general, may contain repetition operators, these intervals can differ in length. The result of a whole query is the set of matches, that is, a set of corpus intervals.

The following examples illustrate some aspects of the query language. Query (2)

(2) `[pos="JJ.*"] [pos="N.*"] "and|or" [pos="N.*"] [pos="IN" & word != "that"]`

returns all corpus intervals which are (adjacent) sequences of an adjective (`JJ`, `JJR`, `JJS`)[6], a noun (`NN`, `NNS`), a conjunction, another noun and finally a preposition or subordinating conjunction (`IN`) which must not be `that` (in the corpus, `that` was often tagged as `IN`, which should be excluded in this query)[7]. When in a condition only the `word` attribute is accessed (together with the equality operator), the brackets can be omitted. So `"and|or"` is just an abbreviation for the complete condition `[word="and|or"]`[8].

Dynamic attributes can be accessed in a simple way:

(3) `"kill.*" []? [pos="N.*" & ishuman(word)]`

As defined in the sample registry file in figure 3, the dynamic attribute `ishuman` requires a string argument and returns an integer value which internally is interpreted as "Yes" if the value is 1, and interpreted as "No" if the value is 0. In query 3, `ishuman` is called with the value of the `word` attribute of the noun. When the query is evaluated, all matches are computed which are a sequence of a word beginning with `kill`, followed by an optional, unspecified word (for example, by), and finally followed by a noun for which the consultation of WordNet gives reason to assume that it may denote a human.

A predefined dynamic attribute is "`f`", which returns the absolute frequency of its argument in the corpus. To search the "most common human beings who are loved", the following query could be formulated:

(4) `"love.*" []? [pos="N.*" & f(word)>10 & ishuman(word)];`

Structural attributes, like sentence boundaries, can be accessed by SGML-like tags:

(5) `[pos="N.*"] [] <s> "She"`

This query returns all corpus intervals where a noun, followed by an arbitrary item (which is to match the full stop or other sentence delimiter) occurs in front of a sentence boundary, followed by the word "She".

Structural attributes like sentence or article boundaries can also be used to limit the search space when repetitions are used. For example, the query

[6] The corpus on which query (2) was run is a part of the Penn Treebank, which has been tagged with the Penn Treebank POS tagset. See [Marcus *et al.*, 1993] for an explanation of the tags.

[7] Query 2 serves to filter concordances which illustrate the problems of adjective scope and PP-attachment within conjoint noun phrases. For a few matching lines, see figure 4.

[8] The condition "and|or" could as well be expressed as `([word="and"]|[word="or"])`, or, abbreviated, as `("and"|"or")`. Whereas the latter two expressions use disjunction on the level of conditions (and have to be grouped by parentheses), the expression used in query (2) uses disjunction on the level of attribute values.

(6) `"president" []* "said"`

would search the two strings "president" and "said" separated by an arbitrary number of non-specified items. In general, only those matches which entirely lie within one sentence will be of interest. This can be achieved by using the `within` construct:

(7) `"president" []* "said" within s;`

Now, the whole match has to lie within the boundaries of one sentence[9]. All structural attributes defined on a corpus can be used as boundary markers (like `<s>`) or in the `within` construct. For example, when the structural attribute `article` was defined on a newspaper corpus, `within article` can be used in queries as well.

An additional, powerful construct of the query language are *label references*, which can be used instead of an attribute value. A condition can be labelled by preceding it with a label name and a colon ("`a:`"), as in (8):

(8) `a:[pos="N.*"] ...`

Then, in a subsequent condition in the same query, an agreement of attribute values can be expressed:

(9) `a:[pos="N.*"] []* [pos="PRP" & num=a.num] within s;`

Here, the value of the num attribute of the personal pronoun (PRP) must be the same as the value of the num attribute at the position the label a refers to, that is, the value of the number attribute of the noun. The whole match must lie within one sentence. Another example which illustrates the power of label references is the following query:

(10) `a:[pos="N.*"] ([]* [word=a.word] ){2} within s;`

This query returns all intervals where the same noun occurs more than tree times within the same sentence.

The query language implements some additional constructs, which cannot described in detail here. For a full description of the query language, its power and a comparison with other corpus query languages, see [Schulze, 1994].

Query results can be saved in files and reloaded and reviewed in later sessions. The logical layer supports subsequent queries on the result of an earlier query, which can greatly reduce the search space and therefore improves efficiency. For example, in a newspaper corpus, one can first extract all articles of a corpus where a certain syntactic construction is used. Afterwards, this "subcorpus" of articles can be analysed by subsequent queries running only on a part of the original corpus. Additionally, set operators are supported, that is, query results can not only be produced by queries, but also by combining results of earlier queries with union, intersection and difference operators. Through this mechanism, it is possible, for example, to intersect the set of sentences generated by a first query with the set of sentences of a second query to get all sentences where the conditions expressed in both queries hold. Although the same result could possibly be produced by a single query as well, set operators are more user-friendly. In our eyes, searching on the results produced by

[9]The number of sentences which may "surround" the matched interval can be expessed with a number following the `within` keyword. "`within 2 s`" therefore allows a two-sentence distance.

earlier queries and the possibility to combine query results to new "subcorpora" supports a successive refinement of queries with the gain of efficiency, and allows a stepwise approach to the solution of complex problems.

The result of a query can be postprocessed by different tools for presentation, frequency counting, additional filters etc. The following section describes two simple presentation tools.

4 Presentation modules

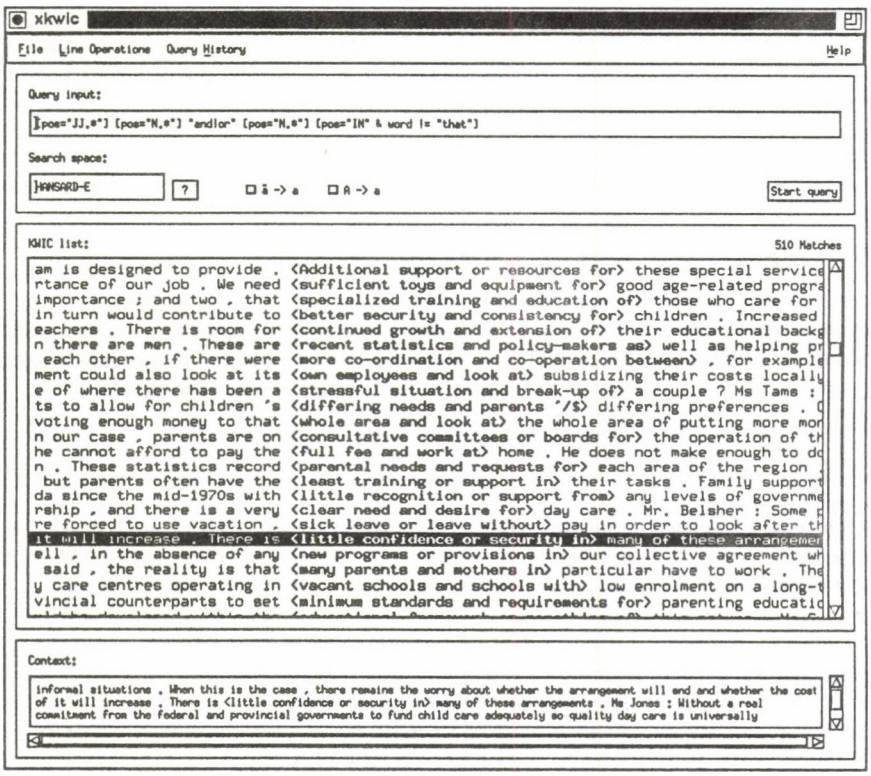


Figure 4: The XKWIC presentation module

A presentation module has the task to display the information returned by a query, suitably formatted for a human user. One instance of such a module is a program called XKWIC which is an X Window System based graphical user interface for displaying key word in context (KWIC) concordances. XKWIC also provides an input area for typing in queries to the logical layer, thus being a general and comfortable interface for corpus work.

Figure 4 shows XKWIC after processing the query displayed in the topmost window

hansard-e: The only way to make and to encourage the responsiveness of child care services to parental and consumer and children 's needs is to encourage competition among those services , to encourage a diversity of services , as indeed exists , so they can reflect the variety of <parental needs and requirements in> this area .

hansard-f: Pour que les garderies tiennent véritablement compte des besoins des parents , des consommateurs et des enfants , il faut favoriser la concurrence entre les services , encourager la diversité , telle qu' elle existe , afin de satisfaire aux exigences et aux besoins nombreux des parents dans ce domaine .

Figure 5: Part of the output of a presentation module which uses alignment information

within the English part of the HANSARD corpus (the same query as shown in example (2) above). The inverted KWIC line is displayed with a larger context in the bottommost window. XKWIC provides functions to adjust the size of the displayed match context, to sort the query result, to delete single or multiple KWIC lines and a function to write (selected) KWIC lines textually to a file. Additionally, XKWIC supports a simple query history: all queries which are entered are kept in a list which can be saved to a file and reloaded in later sessions. An earlier query can then simply be rerun by clicking on the entry in the query history list. XKWIC is described in more detail in [Christ, 1993].

If corpora are aligned (like the HANSARD corpora) and the alignment was defined in the registry file, another presentation module may be used to display both the query result in the source corpus as well as the aligned portion of the target corpus. The same query result displayed in figure 4 then appears as shown in figure 5 (the matching part of the source corpus is surrounded by angle brackets)[10].

5 Further steps

Discussions with users of the query system have shown that it is highly desirable to be able to use parsed corpora in queries. So, one direction of our future work is to design a physical representation of parse trees which allows efficient access and processing and to augment the query language with a construct to refer to this information.

Currently, XKWIC does not support all operations provided by the logical layer, especially the operations on query results (set operations, saving and loading of query results, ...). Therefore, one of our next goals is to integrate the full functionality of the logical layer into a comfortable user interface.

6 Conclusions

The modular architecture of the corpus query system described in this paper has several advantages:

- several knowledge sources can be added to individual corpora. The knowledge they provide can then be used in corpus queries;
- knowledge sources or annotations can be added to a corpus without the necessity of reindexing existing data;

[10] It would be possible to integrate the second module into XKWIC, but this hasn't yet been done.

- through a flexible data model, the information necessary to evaluate the query may be derived from different sources, can be computed at query evaluation time or can be gathered from remote computers;
- through the separation of storage, evaluation and presentation tasks into different modules, the whole system can be adapted to different usage situations.

The flexibility achieved by this architecture, together with the power of the query language, provide the linguist or lexicographer with an extensible and comfortable corpus workbench which allows the querying of corpora with much more precision than within frameworks based only on the corpus text. This leads to more specific queries and results, reducing the amount of data which has to be browsed manually.

References

[Baayen *et al.*, 1993] R. H. Baayen, R. Piepenbrock, and H. van Rijn. The CELEX Lexical Database (CD-ROM). Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA, 1993.

[Christ, 1993] Oliver Christ. *The Xkwic User Manual.* Institut für maschinelle Sprachverarbeitung, Universität Stuttgart, 1993.

[Christ, 1994] Oliver Christ. *The IMS Corpus Workbench Technical Manual.* Institut für maschinelle Sprachverarbeitung, Universität Stuttgart, 1994.

[Marcus *et al.*, 1993] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330, June 1993.

[Miller *et al.*, 1993] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Introduction to WordNet: An on-line lexical database. Technical report, Cognitive Science Laboratory, Princeton University, 1993.

[Schulze, 1994] Bruno M. Schulze. Entwurf und Implementierung eines Anfragesystems für Textcorpora. Master's thesis, Diplomarbeit Nr. 1059, Institut für maschinelle Sprachverarbeitung (IMS) and Institut für Informatik, Universität Stuttgart, January 1994. (In German).

I Can't See the Sense in a Large Corpus

JEREMY CLEAR

Abstract

This paper deals with the particular issue of sense discrimination, which is widely regarded as the central concern of lexicography. Recognising the need for greater sophistication in the software tools that are available to lexicographers for working with a large computerised corpus of English, I outline several approaches to the problem of automatic sense discrimination. An emerging feature of these approaches is that collocation is a powerful organising principle in English. I then describe work in progress on a program to search for particular senses of words. The algorithm makes use of statistical collocational data drawn from 170 million words from 'The Bank of English'. The results obtained so far have been very good and these are summarised and discussed.

1. Corpus tools for corpus lexicography

The COBUILD project, carried out within the School of English at Birmingham University during the 1980s and funded by Collins Publishers, established the importance of using large quantities of real text data in the form of a corpus as the foundation for modern lexicography. In 1987 when the *Collins COBUILD English Language Dictionary* (CCELD) was published, the corpus stood at 20 million words of text drawn from a wide variety of sources. Since that time, COBUILD has been set up as a division of a new HarperCollins Publishers, many innovative corpus-based reference books have been published by the team, the corpus is now known as 'The Bank of English' and the volume and variety of text has continued to grow. At the beginning of 1994 we indexed a 170 million word corpus for the day-to-day lexicography within the group and we expect to set up and index 200 million words by the summer of this year.

The availability of a corpus of this size opens up exciting opportunities for linguistic analysis—statistically-based study begins to show interesting and reliable results, for instance—but it also brings problems. One obvious side-effect of using such a large corpus is that if lexicographers are to describe comprehensively the English which is evidenced there they will require ever more sophisticated software to assist with the sorting, sifting and evaluation of the mass of data. Lexicographers and linguists at COBUILD continue to work with KWIC concordances of words and phrases as the basic research tool, but it is clear that for many words in the central core of English vocabulary items the sheer frequency of

occurrence of these words makes it very time-consuming to carry out an analysis of concordances without further software assistance. We have used an automatic word-class tagging program over the current 170m word corpus and our corpus retrieval software makes use of this annotation to enable fairly complex queries to be handled. In the (common) event that a word being compiled by a lexicographer has a raw frequency in the corpus of many hundreds or thousands, the ability to be able to retrieve or categorize the instances by word-class is often very helpful in discriminating basic word senses. Such discrimination can be based on a simple variation in primary word-class as with the word *collect*, for example, where the meaning 'a short prayer said during some Christian church services' is associated with the noun, and the sense relating to call charges for telephones will be either adjectival or adverbial. Lexicographers at COBUILD are already able to make use of the word-class tags to speed their work and we are working towards further improvements in software which will enable the word-class tagging information to be more fully exploited in identifying phraseological characteristics which contribute to the meaning profile of lexical items for inclusion in our language reference books. In addition, an important research focus for us over recent years has been the exploration of strategies for assigning instances of words in the corpus to sense categories by program. Our short term aim is to supplement the suite of corpus analysis software with a module which takes the set of citations of a word drawn from the corpus and categorises them into subsets which are primarily semantic. If this operation were sufficiently reliable one could find answers quickly and easily to such questions as:

- out of the set of citations of a given word what is the relative frequency of the different senses of the word?
- which citations out of the set do not appear to match any of a pre-defined set of word sense categories? i.e. are there any potentially *new* uses of this word among these citations?

Such questions can be answered at present only by time-consuming and expensive analysis of citations by skilled lexicographers.

2. Automatic parsing of the corpus

Towards this longer term aim of improved automatic analysis, we have commissioned the Department of Linguistics at the University of Helsinki to parse 200 million words of the Bank of English using their ENG-CG Constraint Grammar parser. Professor Fred Karlsson leads the project at Helsinki and 100m of the corpus has already been parsed. It was expected that a full syntactic parse of the corpus would be valuable as a component of a software suite which could identify primary word senses automatically from context. It is far too early to make any firm predictions about the use of parsed corpus data for automatic sense categorization, but I have some personal reservations about the extent to which surface grammatical analysis, of the kind which can be obtained from the current parsing software, will illuminate the complex filigree of lexical meaning. Traditional approaches to the analysis of word senses tend to assume a close and straightforward relationship between categories of grammar such as Object or Subject of verbs and word meaning. I reach now spontaneously for an introductory textbook on linguistics on the shelf beside me and turn to a chapter on 'lexical and grammatical meaning ' and pick a passage almost at random which reads:

> For example, no satisfactory semantic analysis of the noun *picture* could be given which did not state its syntagmatic relationship with verbs such as *paint* and *draw;*

> conversely, the fact that these verbs may take the noun picture as an 'object of result' is to be stated as part of their meaning.
>
> (Lyons 1968: 440)

Appendix A shows ten randomly selected concordance lines for the noun *picture*, from which it will be seen that the idealized locution 'paint a picture ' does not occur, that the verb of which *picture* is the object is apparent in only 6 of the 10 lines and that those verbs are *make, provide, show, supports, develop* and *had*. The only related instance of painting referred to occurs as a nominal head for which *picture* is a noun modifier. It is hard to see in this instance how Lyons's emphasis on the 'syntagmatic relationship with verbs such as *paint* and *draw* ' can be justified.

3. Sense discrimination using dictionary field labels

One approach which seems to offer much more promise of practical and successful application is to make use of existing machine-readable dictionaries as a database for the retrieval of semantic clues which would enable words in context to be disambiguated. I was inspired by the methodology used and results obtained by Don Walker (1987) and Mike Lesk (1986) working at Bellcore. Essentially the approach is as follows. One can make use of the semantic field labels which have been assigned to different senses of headwords in a dictionary as signals of a particular semantic area. One may collect these field labels from the context surrounding an instance of a polysemous keyword drawn from a corpus, and compare the set of labels thereby obtained with the labels associated with each of the different senses recorded for the keyword itself. This instance of the keyword could then be associated with the sense for which there is the greatest observed correlation with the field labels of the context.

This basic methodology has been applied on two independent occasions to COBUILD data using the underlying relational database from which CCELD was extracted as the knowledge base. I carried out my own experiment using the field labels as semantic clues to assign concordance lines to one of the CCELD sense categories (Clear, 1989) and Andrea Lewis (now Assistant Computer Officer at COBUILD) made this approach the topic of her MSc dissertation (Lewis 1992). Although initially the results of both tests seemed encouraging, we both attributed the crudeness of the sense discrimination that could be achieved to the lack of precision inherent in the field labelling of the knowledge base.

The CCELD database was peculiar in respect of its field label annotations, in that the lexicographers who compiled the database were free to annotate senses with labels of their own choosing and multiple assignments were not only permitted but encouraged. The justification for compiling the database without any prescribed set of semantic field labels was that since there is little rigorous theory of semantic primitives the lexicographers should be allowed to record their analysis of the raw evidence of the corpus in a truly descriptive, not prescriptive, way. The aim was to avoid the danger of misrepresenting the evidence of the corpus by forcing our observations to conform to an arbitrary and unproven set of labels. However, the field labels have not, as originally planned, been subject to a revision which would synthesize the labels into a smaller, more rigorous set. There are just over 9400 different labels used in the database of which 3216 occur once only. Thus the reliability of the matching process used is likely to be significantly affected by the idiosyncrasies of the annotation in the database. Since the number of labels employed is far greater than the 120 primary subject-field codes and 212 subfield codes used in the LDOCE database upon which Walker & Amsler (1985) and Walker (1987) based their experiment, one might expect a greater

degree of semantic discrimination in the procedure I carried out. On the other hand, the probability of achieving a match between any two field labels drawn from a set of many thousands is clearly much smaller, and in this respect one would expect better results if the field labels in the lexical database were thoroughly revised and made consistent. Lewis reports disambiguation success of only 60% on average over her test data. The inadequacy of the field labelling caused us to drop this line of investigation as a primary means of automatic word sense or homograph discrimination.

4. Sense discrimination using a machine-readable thesaurus

Another line of research which caught our attention was that followed by David Yarowsky (Yarowsky 1992), who used machine-readable versions of a printed thesaurus and a large encyclopedia as the knowledge base for homograph separation. The basis for the method is statistical and rests on a simple model which assigns a probability that a word belongs to some Roget category:

$$Pr(w \mid RogetCategory_i)$$

Of course, no large corpus exists in which all the words are already assigned to Roget categories, so the probabilities of the model have to be estimated by some other means. For each of the 1042 major semantic categories of Roget, the words in that set (I will refer to this as the 'primary set') are concordanced in Grolier's Encyclopedia. This yields a much larger set of words (say, the 'secondary set') which co-occur with the primary set and which can simply assumed to belong to the same Roget category as the primary set. In many particular instances this simplifying assumption will be false: Yarowsky notes that the word *crane* is a member of the primary set for Roget category 348 TOOLS/MACHINERY but that some instances of *crane* in Grolier will refer to a bird and not to a piece of machinery. The contexts for the bird sense will therefore be introduced as 'noise' into the secondary set. When this procedure is run over every Roget category, the result will be a database containing estimates of the probability that some word w belongs to some Roget category C.

Using these estimated probabilities, Yarowsky selected test words from new, unseen data along with 100 words of context and scored each context using:

$$\prod_{w \text{ in context}} Pr(w \mid RogetCategory_i)$$

and selecting the Roget category which scored most highly. To avoid calculating the score for all 1042 categories for each test context, only those categories in which the test word itself appears are scored.

Yarowsky makes some statistical adjustment to the calculation of probabilities to deal with the differing frequencies of words from the primary set. Suppose, for example, that out of category 348 the word *drill* is very much more frequent in Grolier than *adze*: in this case the context words around *drill* would show a much greater probability of belonging to category 348 than the context words of *adze*. The concordance data is therefore weighted to reduce the contribution of the concordances for drill and proportionally to increase the data for *adze*. The results cited by Yarowsky are very impressive. He reports 93% correct disambiguation averaged over a set of 12 test words.

Last year, Zoe James implemented Yarowsky's basic algorithm within COBUILD, using the same Roget text, but using the Bank of English instead of Grolier's Encyclopedia.[1] The results were somewhat disappointing. I am not able to report accurate figures for the success of the method as Zoe James implemented it, because firstly our casual and informal evaluations quickly caused us to abandon further work, and second the procedure was constantly being adjusted as we reacted to the almost daily suggestions and comments of Bill Gale of AT&T Bell Laboratories, who helped us with the statistics. In summary, however, the method seemed to us to depend too heavily on the 1042 Roget categories. Where a sense distinction could be clearly and absolutely related to the thesaurus categories, performance was good. But our lexicographers regarded the semantic classifications of Roget as mostly unhelpful in relation to their own perceptions of word sense, and in a few instances bizarre. We did not consider it worth preserving the huge data files which were created during the course of this work, so I do not have detailed material which can be presented to show the results we obtained, but we were concerned about the generality of the Roget categories. For example, the word *badger*, which we used as a test word, appears under the headings ANIMAL/INSECTS (which seems appropriate) and UNPLEASURE (which does not).

5. Collocations and word senses

Though the Roget method was not well suited to our needs, it led us to focus our attention on statistical collocation as a potentially powerful lever on word sense discrimination. Indeed, Yarowsky's paper confirmed what had been already become well established in the minds of many within the COBUILD team—that the simple association of lexical forms which can be detected in a large corpus provides a vital clue to the identification of lexicographic word sense classification. Yarowsky's more recent paper, entitled 'One Sense Per Collocation' (Yarowsky 1993) pursues this line of thinking and reports briefly on experiments carried out to measure the extent to which word senses are associated in an information-theoretic way with particular collocations. He concludes:

> Experiments have shown that for several definitions of sense and for several definitions of collocation, with high probability an ambiguous word has only one sense in a given collocation. . . We also show that this results in high precision sense disambiguation when collocational evidence is available.

What is perhaps startling and novel to a computational linguist, is surely all too obvious to a lexicographer who has made extensive use of a large corpus in their work.

Our most recent work on sense disambiguation has taken up the idea that collocation offers the best prospect for achieving fully automatic sense discrimination. Let me present some of the reasons for our optimism:

> collocation (as defined in the way I describe here) is a fundamental organising principle for English, bearing upon all other more abstract theoretical hypotheses about the structure of the language (e.g. syntax, morphology, semantics, 'deep' grammar),

---

[1] I am very grateful to Bill Gale of AT&T Bell Laboratories, New Jersey, who helped us by providing many valuable suggestions and explanations relating to the statistical calculations in this process.

- the regular practice of lexicography leads us to find meanings more often dispersed across loose phraseological constructs than isolated within individual orthographic words,

- the availability of a reasonably large corpus of 170 million words for computational manipulation has enabled us to use statistical models which have already proven useful in linguistic analysis (e.g. in word-class tagging).

6. A program to discriminate word senses

In an earlier paper (Clear 1993) I considered the use of collocation lists extracted from a corpus and discussed the types of patterning which emerges. My current work has been directed towards using this information to automate the process of identifying particular senses of words. For greater clarity of explanation, let me define some terminology: this will be consistent with terms used by Sinclair (1991) and Clear (1993). The **node** is the word form (a character string) under investigation. A KWIC index of the node is assumed to be available, which is composed of **citations**—one for each instance of the word in the corpus. The **span** for each citation is a variable number of characters or words to the left and right of the node. The individual wordforms within a citation, excluding the keyword, I will refer to as **collocates**.

The procedure we are currently using begins with generating a list of significant collocates (where significance is determined solely on statistical grounds) for a node to be disambiguated. For example, the top of the collocation list for the node *bow* sorted in decreasing order of significance is as follows:

```
Node: bow 3020
wow                 1049      257    16.02
tie                 6054      170    12.97
to               4340863     1008    12.87
a                3752112      847    11.30
and              4101538      888    10.80
street             60288      121    10.24
with             1101447      295     8.32
out               353997      150     8.26
bow                 3020       64     7.95
pressure           25896       70     7.94
ties                5191       57     7.45
arrow               1187       52     7.19
magistrates         2246       50     7.03
his               718514      198     7.02
the              9900535     1647     6.90
arm                10814       49     6.79
her               375931      123     6.41
tied                4871       40     6.22
stern               2389       38     6.11
doors               7135       39     6.09
string              4329       37     5.98
take              118124       64     5.96
wave                6779       37     5.93
across             35126       44     5.90
starboard            595       34     5.82
```

The four columns of data are:

- the collocate
- the overall frequency of this collocate within the corpus
- the frequency of co-occurrence of this collocate with the node
- the significance value of this collocation

In this example, as very often occurs when studying collocation in a large corpus, certain collocates in the list are clearly associated with different senses of the node. Since we can distinguish verb instances of the node from nouns, I will ignore the verb senses (and the phrasal verbs, such as *bow out*, which we see here). When we see the collocate *tie*, for example, we recognise the 'knot' sense of *bow*: below I tabulate the correspondences.

Sense Category	Collocates
a type of knot	tie, ties, tied
a weapon	arrow, string
the front of a boat or ship	stern, doors, wave, starboard
a deferential gesture	take
**famous law court	street, magistrates
**doggy noise	wow

Table 1.

It is typical of this sort of analysis that some collocations are stereotyped to the extent that they form multi-word lexical units (*bow wow* and *Bow Street Magistrates Court*) whose meanings are not really discrete senses of *bow* at all, and these I have marked with asterisks. I will reconsider these fixed collocations later, and focus for now only on the senses which one would expect to find listed in a dictionary under the simple headword. The collocate *string* might signal another sense of the node: the thing you draw across the strings of a violin or other stringed musical instrument. These associations are impressionistic and informal, of course. This manual stage of analysis, however, can be very quickly performed by lexicographers and with a very high level of consensus. If we are uncertain which of two possible senses a collocate might signal, we can very easily consult the citations for the particular node+collocate pair to resolve the matter or else simply ignore this collocate as a poor discriminator.

The collocates in the right-hand column of Table 1 seem to be good discriminators of word sense. The daily work of corpus-based lexicography has provided COBUILD with overwhelming evidence of such a correspondence, and Yarowsky (1993) adduces experimental results which tend to confirm it. These collocates seemed to be a better starting point than the words grouped with Roget categories for automatic sense discrimination. So we can now partition the collocates into *two* sets: the first relating to a sense which the computer is to identify, and the second being collocates relating to all other senses. The first set we term **clues** and the second set we term **antis**. I have written a program which takes as input two short lists of clues and antis and a set of citations from which instances of a particular sense are to be retrieved. I have been running the program with around four clues and eight antis over test words which have between two and six senses listed in CCELD, so that the manual task of selecting discriminating collocates is trivial.

Now the program calculates across the whole corpus a significance score for all collocates of each of the clues. This step is analogous to the concordancing of Grolier which is carried out in Yarowsky's method. The aim is to obtain a list of words (let us call them **discriminators**)

each with a score, the value of which indicates the strength of association of the discriminator with the word sense of the node which we are attempting to identify. We will then look for these discriminators in the context of citations of the node and pick out citations which have a good number of discriminators (or a few strongly associated ones). To improve performance, we also extract from the corpus collocates of the antis and we add these to the list of discriminators, having first inverted their significance scores. This will perhaps be clearer if I work through the example of *bow*. Suppose we wish to identify the nautical sense of *bow*. We select the clues *stern, ship, starboard* and *port* from the initial list of collocates.[2] We might use the other words shown in Table 1 as antis. The program now obtains collocates of the clues, and for *starboard* we get a list which begins thus:

Node: starboard 595			
companionway	104	6	11.05
settee	340	12	10.34
shrouds	123	4	10.22
buoys	183	5	9.97
astern	198	5	9.86
stowage	179	4	9.68
undercarriage	180	4	9.68
galley	818	12	9.08
berth	688	10	9.06
rudder	447	6	8.95
starboard	595	8	8.95
aft	678	9	8.93
tack	837	10	8.78

These words are discriminators and each has a significance score (in the fourth column) which is a measure of the strength of association of the word with *starboard*. What we hope is that these words will also be associated with the nautical sense of *bow*. This list of discriminators will be lengthy (in proportion to the overall frequency of the clue word) and will include some words which have little or no statistical association with *starboard*, but we are not concerned about these. We now supplement this list of discriminators by taking a word such as *arrow*:

Node: arrow 1187			
debreu	6	6	14.17
bowditch	9	6	13.59
fromstein	26	5	11.79
stillwaters	289	13	9.70
quiver	160	5	9.17
clowes	167	5	9.11
amis	472	11	8.75
bournemouth	1567	35	8.69
837	358	7	8.49
bow	3020	52	8.31
firepower	355	6	8.28
flaming	384	5	7.91
pierced	468	6	7.88
berry	1392	17	7.81
blue	21047	253	7.79
arrow	1187	14	7.76
guinness	1333	14	7.60
plc	1831	18	7.50
barlow	637	5	7.18
spear	690	5	7.06

This list seems to be badly skewed by references to a financial scandal in the UK concerning a company called Blue Arrow, but there are a few useful words in the list such as *quiver, pierced,* and *spear*. Of course, these are negative discriminators because their appearance in the context of some citation for *bow* would be a strong indicator that the citation does not exemplify the nautical sense. For this reason we turn the large positive significance scores into negative scores and vice versa, before adding these words to the growing list of

[2] The list printed here is a small fraction of the full list of significant collocates of *bow* and the clues selected are taken from further down the list.

discriminators. All this processing is automatic and the list of discriminators generated by each clue or anti word will in practice contain thousands of words. In fact we obtain over 13,000 discriminators from these clues, and over 52,000 discriminators which collocate with the following antis which I selected:

```
street tie tied string arrow court legs strings shot final
```

The program then reads citations for the node to be disambiguated and checks each word within a span of 512 characters of the node, looking for a match with the list of discriminators. Whenever a match is found, the significance value of the discriminator is added to a cumulative score for the citation. Finally the citations are ordered by their overall score and output. Appendix B shows the resulting output for 100 randomly selected citations of *bow* as a noun.

7. Preliminary results

The sense discrimination program has been run over sets of 100 citations for different nodes. Citations which are assigned an overall positive score by the program are those which the program considers to be instances of the required sense, and citations with a negative score are deemed not to be. All results thus far obtained have been subject to human reading and checking, and a contingency table is completed for each test unit of 100 citations.

	Human Yes	Human No
Computer Yes	*A*	*B*
Computer No	*C*	*D*
TOTAL	*E*	*F*

Thus for the test data for *bow* shown in Appendix A the results are:

A=25	*B*=1
C=2	*D*=72
E=27	*F*=73

If we treat the task as an information retrieval task, then we could express the results in terms of the conventional **precision** and **recall** measures used in the Information Retrieval literature, where precision is the percentage of relevant items out of the total retrieved items (in this case 25/26 = 96%) and recall is the percentage of all relevant items which were retrieved (in this case 25/27 = 92%). Typically the information retrieval problem is that precision can be improved only with a reduction in recall or vice versa. In the extreme case, the computer could simply mark all 100 test citations as instances of the required sense and thereby achieve 100% recall, but precision would drop according to the frequency of the sense to be identified. Since for our purposes we do not want to make an *a priori* decision about the relative importance of precision and recall, I will follow Gale, Church & Yarowsky (1992) and refer to percentage correct:

$$\left(\frac{A+D}{E+F} \right) * 100$$

which in the case of *bow* is 97%.

Here are the results from other test words we have evaluated:

RESIGNED

'give up a job, position, etc.'

correct = 92%

86	7
1	6
87	13

DIALOGUE

'political negotiation, talks'

correct = 96%

63	0
4	33
67	33

PLOT

'storyline; sequence of events'

correct = 84%

36	2
14	48
50	50

HARBOUR

'conceal (germs, virus, etc.)'

correct = 97%

23	2
1	74
24	76

These bare figures are clearly encouraging, but there is a considerable amount of work still required to investigate fully the performance of the algorithm over a large number of test words and to assess the effects of varying the details of the procedure. Development of the methodology is proceeding rapidly—the results reported here were produced very

recently—but I would like to consider here some of the issues raised in the course of processing which are likely to be the focus of closer analysis.

8. Discussion of results and methodology

8.1 Selection of clues and antis

For the purposes we envisage at COBUILD, the manual selection of clues and antis is no significant problem, but clearly it would be interesting and valuable to explore the possibility of automating this step. At present the manual selection has some justification in that the lexicographer is able to make the decision about which senses are to be isolated. For example, in the case of the word *harbour* as a verb, there are just two senses recorded in CCELD (one for harbouring emotions, the other for harbouring terrorists or outlaws) and the 'germ, disease, bacteria' sense is omitted. A computer search for this latter sense shows that it accounts for about one-quarter of the verb instances. The lexicographer may decide after corpus investigation that two senses already recorded in a dictionary might be better conflated into a more general account of the semantics of the headword. They may then select clue and anti words according to their particular requirements. The selection of clues takes only a few minutes.

We are experimenting with automating the selection of clues and one possibility is to collate the list of significant collocates of the node with the set of words which appear in the full-sentence definitions and in the examples in CCELD and to select as clues words which are common to both. It is the characteristic COBUILD defining style, in which the typical co-text of a word is elaborated within the definition, which makes this approach plausible. In effect the compiler of the CCELD entry has already carried out a selection of clue words and incorporated them into the definition and the examples.

There is a complication with the test word *harbour*. The noun instances in the Bank of English number 2995 compared with 235 verb ones. If collocations of the wordform *harbour* are extracted for the purpose of selecting clues, the noun uses are so dominant that very few relevant collocates achieve a high significance level. In this case it is necessary to carry out the initial generation of a collocate list only over the verb instances, and we anticipate that this refinement will be introduced to the software in due course.

There are some desiderata for the selection of good clues and antis: the words should be frequent enough to generate significant collocates for use as discriminators, but on the other hand not so frequent that they behave more like grammatical function words than fully lexical items. In general, the very high frequency words of a corpus will be grammatical or highly polysemous and in either case unlikely to yield good discriminators.

8.2 Discriminators

After I had made various adjustments to the software and evaluated the effects, it appeared that the list of discriminators must be large in order to achieve good results. The failure of the method which used field labels from the CCELD database is primarily due to the paucity of information available from a set of just a few thousand words. This new method generates lists of discriminators running to tens or even hundreds of thousands. An interesting question for the improvement of the method is to what extent noise (in the information theoretic sense) can be tolerated in the discriminators list. This is relevant to the selection of clues and antis, since a bad choice of clues would result in the introduction of irrelevant items among the

discriminators. However, I am intrigued that our intuitive notions of the 'relevance' of other words to the sense we wish to identify are somewhat narrow. If the matching discriminators for a particular citation under test are displayed along with their significance scores, it is difficult for the human analyst to find any intuitive confirmation of the usefulness of most of the discriminators. It seems that the gross cumulative effect is not predictable by analysis of the individual matches of context word and discriminator. For example, the word *once* occurs as a negative discriminator for *harbour* in the 'germs, virus, etc.' sense, but I have no idea why this word should be a useful indicator, nor indeed whether it is simply noise.

8.3 Local stereotyping

The results I have obtained so far are best when there are very clearly demarcated subject domains in which the different senses of the node characteristically appear. Many words in the central core of English vocabulary, however, combine with varying degrees of fixedness to create modified meanings as phraseological constructs. Of course, it is precisely this tendency which makes the algorithm I am using useful. But where the associations (which I term **stereotyping**) are very localised to within a word or two of the node, the algorithm start to perform less well. There are several manifestations of stereotyping which I have found to degrade performance of the program. First, phrasal verbs cause classification errors, since they will appear within the citations for the test node but sometimes with a meaning which is unrelated to the primary senses of the node in isolation. One can deal with this by artificially treating the phrasal verb as if it were simply another sense of the unitary node. So, for example, the program can search for the word *drown* with the sense 'cover one noise with a louder one', when in fact it is the phrasal verb *drown out* which has this meaning. Compounding (*heavy metal* = 'a type of rock music') causes the algorithm to miss instances where the node forms a fixed unit with an entirely different meaning and less fixed idioms such as *drown your sorrows* need special attention. Arbitrary collocations are perhaps the most problematic of the stereotyped patterns, since the tradition of lexicography does not accord the status of discrete entries to such combinations as *light snack/meal, heavy rain, heavy trading* on the financial markets, *light dusting* of icing sugar on a cake and so on. These stereotypes cannot easily be subsumed within the core meanings of *light* and *heavy* found in most dictionaries. To a large extent this is not a problem for our automatic sense discrimination program, but for lexicography. One of the visible effects of the corpus analysis which underpinned the compilation of CCELD is the tendency to treat arbitrary collocations as discrete numbered senses, in a deliberate move to extend the traditional bounds of lexicographic practice. An appropriate strategy for dealing with this phenomenon may be to extract such instances as a preliminary step (using the collocational statistics to identify them) and to select clues and antis only after the highly stereotyped combinations have been removed from the raw data.

9. Conclusion

We are very much encouraged by the results we have obtained to date. This methodology has a firm basis in collocation, which I regard as a powerful organising principle in English, and benefits from the lessons learned from closely related experiments in word sense discrimination. The results compare favourably with those reported in the literature of automatic sense discrimination experiments. We are sufficiently confident of the accuracy of the software to introduce it within the routine corpus analysis tools available to lexicographers. We will then be able to evaluate its performance in a production environment over a very wide range of lexical items and refine and tune the program in response to

feedback about its failures. The software is reasonably efficient as it now stands: once the clues and antis have been selected, it takes about 2 minutes to process several thousand citations of a node word. The program can be used iteratively over a large set of citations in order to peel off successively the different senses of the node in accordance with the lexicographers interpretation. In the longer term we expect to automate the selection of clues and antis with little or no degradation in the success rate for the program, with the clear prospect of being able to reduce significantly the amount of tedious human classification of concordance lines which is still a major part of lexicographers' work. The numerical scoring which the program generates enables borderline instances, or at least those which the computer cannot securely classify, to be presented to the lexicographer for detailed analysis.

References

Clear, J. H. (1989) 'An experiment in automatic word sense identification', unpublished working paper, COBUILD. (Copies available from the author.)

Clear, J. H. (1993) 'From Firth principles: computational tools for the study of collocation' in M. Baker, G. Francis and E. Tognini-Bonelli (eds.) *Text and Technology: In Honour of John Sinclair*, John Benjamins.

Gale, W., K. Church and D. Yarowsky (1992) 'Work on statistical methods for word sense disambiguation' in Working Notes for AAAI Fall Symposium on Probabilistic Approaches to Natural Language.

Lesk, M. (1986) 'Automatic sense disambiguation: how to tell a pine cone from an ice cream cone,' *Proceedings of the 1986 SIGDOC Conference*, Association for Computing Machinery.

Lewis, A. E. (1992) 'Lexical disambiguation using field labels in the COBUILD database', MSc dissertation, Cognitive Science Research Centre, University of Birmingham.

Lyons, J. (1968) *Introduction to Theoretical Linguistics*, Cambridge University Press.

Sinclair, J. McH. (1991) *Corpus, Concordance, Collocation*, Oxford University Press.

Walker, D. (1987) 'Knowledge resource tools for accessing large text files' in S. Nirenburg (ed.) *Machine Translation: Theoretical and Methodological Issues*, Cambridge University Press.

Walker, D. and R. Amsler, (1985) 'The use of machine readable dictionaries in sublanguage analysis' in R. Grishman and R. Kittredge (eds.) *Sublanguage: Description and Processing*, Lawrence Erlbaum.

Yarowsky, D. (1992) 'Word-sense disambiguation using statistical models of Roget's categories trained on large corpora' in *Proceedings, COLING-92*, Nantes, France.

Yarowsky, D. (1993) 'One sense per collocation' in *Proceedings, ARPA Human Language Technology Workshop*, Princeton, New Jersey.

Appendix A

10 Randomly Selected Concordance Lines for *picture*

```
<        the metal to make boxes,chandeliers, sconces and picture and mirror frames for their homes. Unlike the
<their improvisations, toymakers, town square performers, picture painters, music players, festival dancers, sculptors
<     In each case, Humphrey Bogart took Raft's place. The picture that Raft did consent to make, Manpower,
< El Nio events. The next task was to provide a dynamical picture of that process, one that could use observations of a
<But this section isn't concerned with that aspect of the picture. I want to draw your attention to the fact that the
<  to her, 'Ma'am, unless you're just trying to show me a picture of the late president, you better put that back,
<      building block of life, the DNA molecule,supports a picture of mutations weeded out by competition and passed
<     activities. She was beginning to develop a realistic picture of how she would like to be. She remained ambivalent
<  third he sees the ox. The series proceeds to the ninth picture in which the hero tames the ox, forges a peaceful
<   and expense each possible client until it had a clear picture of that business's needs. It then designed its
```


Appendix B

Sample Program Output for *bow*

```
710.73: ttack which blew the bow off. The ship sank swiftly by the bow as Martlets, ranged at the stern, broke their lashings an
538.76: pletely refurbished at a cost of 5M # Gone are the bulbous bow structures for pulling submarine cable on board # The ste
509.69: ppears all of the ship's nine oil tanks were damaged # The bow sank into the sea this morning # Authorities decided to l
411.22: looking up at the sails billowing about, then watching the bow cutting through sea below. It was an experience to savour
403.80: achts must be prepared to be positioned 'centre lock' by a bow and stern line on each side, hence the requirement for fo
387.48: he pontoon from any further aft. Holding the boat with the bow close enough to the berth for a youngster to negotiate th
384.74: d box. <LTH> So the 950's final shape is a compromise. The bow is fine at the waterline but flares out markedly toward t
357.86: re of with a hefty bow plank housing the rollers. The main bow anchor self-stows here. All deck gear is substantial and
356.47:  my stern warp parted and a horrible grinding noise as the bow gouged along the harbour wall before parting the bow line
301.83: the Atlantic then travels the length of the boat until the bow is finally dropped into the following trough. The timbers
298.64: laughed and swung himself on board, working his way to the bow and the gong. The anchorman waited there, too, but instea
290.05: apsized on March the sixth, 1987, after it sailed with its bow doors open # Graeme mclagan reports from the Old Bailey:D
273.59: verboard!" was heard. A male passenger had fallen from the bow. It was dark by now and nothing could be seen in the wate
176.31: ails carve swaying black shapes against the stars, and the bow wave foaming phosphorescent. But I can only tell you that
```

130.60: he bow is well fendered, like this, and get someone in the bow to call back the approximate distances. Stand high to one
122.45: ery now and then it makes a sound like grinding molars. On bow lookout I am told by Gregg, and then by Joan, and then re
122.31: n <CES> twenty <CES> thirty seconds.<t> 2.20 am # With the bow well under water, the remaining hull slipped downwards wi
111.99: ike Thornton asks how the bow tie became known as a dickie bow. The answer lies in the Dutch verb 'dekken", to cover. <t
110.35: e captain. The Three Investigators and Jeremy stood in the bow of the ship as it ploughed through the water of the cove
97.36: and tie the webbing to the rowlocks. You can then lift the bow and pull the boat comfortably. The theory worked well but
70.45: de Kersauson's Charal crashed into the ice and smashed the bow off the starboard float of the 90ft trimaran. The crew ar
67.64: ith just two adults was made easier by having both the 30m bow and 20m stern line controlled by one person in the cockpi
66.64: kay. So it makes sense. And again we see <ZF1> this little bow wave <ZF0> this little bow wave of compression just befor
52.96: e to mind. She lined up several landmarks on the trawler's bow then, with heart pounding, zigzagged her way towards the
28.20: wrie Smith, skipper of Fortuna, a maxi with a conventional bow, dismisses this as arrogance.<LTH> If you read the rule y
13.64: e sun on a sapphire ocean. The splash of waves against the bow..the glint of flying fish and the flash of sunshine on th
-18.48: kout on the night # It was a bright,moonlit night when the Bow Belle collided with the Marchioness which was carrying a
-46.73: <FCH> Diana and Artemis <FCH> Divine Artemis of the golden bow, under whose protection live stags and hinds, bears and w
-57.73: still remember the fearful guilty thrill of watching Clara Bow oozing It" as I sank my teeth into a sumptuous peppermint
-60.60: a saw when he was escorting her to Cengarn. Does he have a bow with him # Not that I know of. Does he know how to use on
-64.08: work out. Chopping your wood, sawing it into pieces with a bow saw <KUA> living like that takes so much sheer bloody toi
-66.70: bigger waves,rather than over them. <LTH> In practice the bow chucks them aside without lopping the tops off, and the i
-68.99: M0X> Got <ZGY> <M0X> Yeah four incipient [pause] houses on Bow Street.<M0X> <ZGY> them.<M0X> Four hundred quid.<M0X> Yea
-85.24: is a long bow for but to be drawn? And our phrase the long bow" itself comes from the great bow of Philoctetes, one of t
-88.31: out the rather autocratic director-general Amadou Mahtar M'Bow when I was first appointed, but that was chiefly a proble
-102.08: t having seen him for so long.<t> He gave that exaggerated bow which always irritated me. If it is not Miss Ann Alice he
-108.63: end to press for clarification," Rifkind has told the Tory Bow Group. 'The explanation given by Germany is that it is to
-117.26: ved to the National Railway Museum. <LTH> Class 73/0 'jas' bow out: From the start of the summer timetable the class 73/
-118.09: man pyramid whose base wanders to the footlights to take a bow, cheap disappearing acts, a demure bump-and-grind from Na
-119.43: ing parliament as MP for Rushcliffe in 1970.<LTH> He was a Bow Grouper, not only liberal on social issues but economical
-123.22: also I wasn't a manager, if you see what I mean. But with Bow Wow Wow I was becoming too <FCH> much <FCH> a manager,and
-127.62: . Avoid. <LTH> PETE FRAMES'S ROCK FAMILY TREES # Omnibus), bow in a complete anthology, are strongly recommended for the
-131.12: used for entertaining heads of state. Ronald reagan took a bow there, as did Lech Walesa, and Mikhail Gorbachov. <LTH> S
-132.61: er. The latest move by the Bush administration is partly a bow to political reality. The Democrats on the Senate Foreign
-133.46: and groups to send their details to them at 3 Coborn Road, Bow, London, E3 2DA. <LTH> <LHH> New race for Heathrow <LTH>
-135.92: op listening to the animals, you know # Oh, yeah, that's a bow-wow.'And I hope that this book doesn't undermine a lot of
-137.12: n # <FCH> Animal:<FCH> Horse.<t> <FCH> Hsiu 26 Chang Drawn Bow; Extending,spreading <FCH> 6 stars E.E: (i) 18.05. (ii) 1
-138.59: her father was a real East-ender, born within the sound of Bow Bells, she was brought up on a farm in Devon. <LTH> I was
-142.90: it more elegant?", and gave a mock Renaissance courtier's bow. The poor man laughed nervously, but it was clearly as if
-149.62: LA LATE THAN NEVER <LTH> Regarding the recent WSF piece on Bow Wow Wow's ANNABELLA LWIN, it may be of interest that Anna
-159.61: rst prize of Pounds 6,000 and a Garner Wilson gold-mounted bow. <t> His cool, calm and collected performance of Brahms,
-160.42: gown who with admirable grace returned them with a polite bow and a short exchange of words with the men of the family.
-169.21: the question. Pasquaanti would simply remind them all that Bow-legs was a Navajo # thereby explaining the gap in academi
-177.40: d normally bash someone like that. <CQ0> <t> George joined Bow Wow Wow for the first three dates of their first U.K. tou
-179.58: tralian soprano, Dame Joan Sutherland, has taken her final bow at the Royal Opera House in London's Covent Garden, nearl
-185.15: stands sentinel to the bridge, and on the other a pretty, bow-fronted chemist's shop. The heart of Dunkeld is a small '
-196.66: own doubts about the cassette-only format.Considering that Bow Wow Wow's first product was to be released at nearly the
-197.15: t dare to greet him with anything more than an abbreviated bow.<t> Reviewing the restless, excited hall, Man was mildly
-206.22: ! Mr Goodwin # <t> The selfsame," I answered with a slight bow and an earnest smile. 'And you are Maria Radovich, I pres
-211.91: from Barrie's company, The Imperial Bathroom Company). The bow-topped mirror above the basin, the bath side and WC seat
-225.15: ry useful after I set up on my own, after a spell with the Bow Street runners # <t> Aye," said Stephen, 'I am sure it wo
-228.17: e were so many Feydeau farces about (Phoenix and Stratford Bow) and 'they're all so predictable." They're all desperate
-235.40: Publications Act. <LTH> The newsagent was due to appear at Bow Street Magistrates' Court on March 23rd to face the charg
-243.71: n or Des on 0204-390 445 or write to Bolton Gay Centre, 11 Bow Street, bl1 2EQ. <LTH> YOUNG GAY-LESBIAN INITIATIVE BOLTO
-256.45: arty were in a relaxed mood when they set off with the old bow-legged Joe, who was in constant pain from a back problem.
-258.66: llroom entrance and acknowledged the applause with a small bow. He was not a tall man, but something about his confidenc
-263.57: ide his mother he stood, his strong hands holding a strong bow and many sharp iron-tipped arrows.<t> When the vision had

-263.77: many of the country's most interesting artists, including Bow Gamelan, Alastair maclennan and Neil Bartlett. Third Eye
-273.81: oceedings. <t> Later Travers refused to comment as he left Bow Street clutching his side in pain and still wearing his s
-275.93: own <KPD> 15.99). when the New Police replaced the corrupt Bow Street Runners in 1829, they carried on the old tradition
-280.67: gic Flute in the Freemason's Hall and Trial by Jury at the Bow Street Magistrate's Court. <LTH> <LHH> PERSONALS <LTH> Wr
-283.28: them, from the slightest jibsheets to the three-inch-thick bow-line on the <FCH> Range Sentinel, <FCH> was a single, uni
-284.17: 29.50, plus <KPD> 2.50 p&p, from Round the World, 82 West Bow, Edinburgh eh1 2HH; tel: 031-225 7086. <LTH> <LHH> SEATS
-298.78: ssue.Schorr: Shareen, Jeffrey talks about shots across the bow which reminds me that we get Iraqi surface-to-surface mis
-301.00: olence must be met with punishment.' <t> mcclean, 42, from Bow, South London, admitted assault causing actual bodily har
-306.14: rch a choir of country kidsin clean blue jeans and clip-on bow ties sang # The first noel the angels did say was to cert
-312.07: 1 tsp salt <LTH> 1 tbsp olive oil <LTH> 350g (12oz) dried bow pasta <LTH> 150ml (quarter pint) low-fat fromage frais <L
-320.96: He is one of those players who has an extra string to his bow. He can perform comfortbaly both in midfield and at the b
-354.87: e dim distance. And at the end the ritual Islamic embrace, bow and prostration carried that extra air of triumphalism. <
-355.02: leading guilty to public order offences. <LTH> The case at Bow Street Magistrate's Court was brought under Section 5 of
-358.04: r representative sockets to full depth, then place the arm bow on to them. The arm bow should rest easy at the prescribe
-385.27: n said this week. You could imagine him in dinner suit and bow-tie expressing Auntie's disappointment.There was, accordi
-385.95: broad-brimmed hats, but I felt that the Duchess of Kent's bow and veil was a very original idea and could start a trend
-387.36: er back stick socket positions to the underside of the top bow and drill these sockets carefully as in Fig. 8. It will a
-390.34: Michael Stoute certainly has a strong second string to his bow because she accounted for Shirley Valentine, among others
-391.90: ie, and Madame Lucie cut it down on the spot into a narrow bow. The other ladies were transfixed, hands clasped in wonde
-402.82: piracy before being granted unconditional bail at London's Bow Street Magistrates Court. <t> Magistrates cleared a fourt
-411.99: hey called Beck <CQ1> Mad Dog. <CQ0> Although he sported a bow tie and horn-rim glasses, Beck more closely resembled a c
-422.27: lack trousers, those at the top add wing collars and white bow ties, to which the most privileged of all -members of 'Po
-450.75: nd with a voice so low it was inaudible, he stood before a Bow Street magistrate in a dirty shirt and torn, sagging jean
-465.26: n Sec of the Barchester Rosemary Clooney Fan Club. I was a Bow Groupie when Reggie Bevins was thinking of locking up Sir
-528.66: but he took her hand and led her to the bed. She undid his bow tie and took the links out of his cuffs and folded Klaus'
-540.09: ose). Average weight:<LTH> 13st 12 1/4 lb.<LTH> CAMBRIDGE: Bow *D E Bangert (Deutschhaus Gymnasium Wurzburg, Univ of Wur
-570.35: e is threaded through a small hole in the other end of the bow, it is pulled through a guide loop, then through the tuni
-610.24: bell-bottoms, these enormous lapels and a velvet butterfly bow tie # And he's wearing these super-high, platform shoes h
-640.58: elf # in his blue suit pants, button-down shirt, signature bow tie, cashmere cardigan, and long hair, he comes across as
-733.29: beautifully cut tails with a white silk shirt and a black bow tie. Since it was impossible for Mick to wear anything tr
-821.58: l figure looking back at him as he gave a final tug to his bow tie, adjusted the lapels of his white tuxedo, and smoothe
-898.76: /03/89 </dt> <pb> WALL STREET JOURNAL (J) </pb> <co> MCD S BOW GOVMT FRE+ G.BAS EUROP JAPAN UTX </co> <in> BOND MARKET N
-3565.63: brother, hunter of dawn,prince of the woods, expert at the bow and arrow, I welcome you for the joy your presence brings

A Parallel Approach to Lexicon Design

MARKUS DUDA

Abstract

To communicate with a computer in spoken language is an unattained challenge of Artificial Intelligence (AI) and Computational Linguistics. To solve such problems linguistic knowledge has to be combined with programming methods of AI and modern computer architectures. We will show how the complexity of linguistic processes can be handled by taking advantage of parallel architectures. In particular, speech systems where most lexicon queries are extremely underspecified suffer from the problem that the access to the lexicon module turns out to be a bottleneck.

We introduce the *search problem* over a given lexicon and compute its time complexity for two different encodings. With the help of a space consuming encoding we define a total order over a lexicon, and, having a total order, logarithmic time becomes valid for the complexity of sequential lexicon search. Next, we will speed up the search by parallelisation, making use of the *paracomputer*. Last, we describe a practical approach to the parallelisation of a lexicon module with the aim to maximize the throughput.

1 Introduction

The most common form of linguistic data representation is the **feature structure**. So, for linguistic applications in computer science the lexicon is an abstract data type over sets of feature structures with minimally one function for the search. The search function selects zero, one or more elements from the lexicon set which fit a search pattern. The example gives the lexicon description of the German word form *Weibern (dative plural form of female):*

$$\left[\begin{array}{ll} \text{LEX} & : \textit{weib} \\ \text{ORTH} & : \textit{"Weibern"} \\ \text{SYNSEM} & : \left[\begin{array}{ll} \text{LOCAL} & : \left[\begin{array}{ll} \text{CAT} & : \left[\begin{array}{ll} \text{HEAD} & : \left[\begin{array}{ll} \text{NFORM} & : \textit{nform} \\ \text{CASE} & : \textit{dat} \end{array}\right] \\ \text{LEX} & : + \end{array}\right] \\ \text{CONTENT} & : \left[\begin{array}{ll} \text{INDEX} & : \boxed{1} \left[\begin{array}{ll} \text{NUM} & : \textit{plu} \\ \text{PERS} & : 3 \\ \text{GEND} & : \textit{neut} \end{array}\right] \\ \text{RESTR} & : \boxed{2} \left[\begin{array}{ll} \text{REL} & : \textit{weib} \\ \text{ARG1} & : \boxed{1} \end{array}\right] \end{array}\right] \\ \text{CONTEXT} & : \boxed{2} \end{array}\right] \\ \text{NLOC} & : \textit{nonlocal} \end{array}\right] \\ \text{QSTORE} & : \textit{list} \end{array}\right]$$

A possible search pattern for this entry is

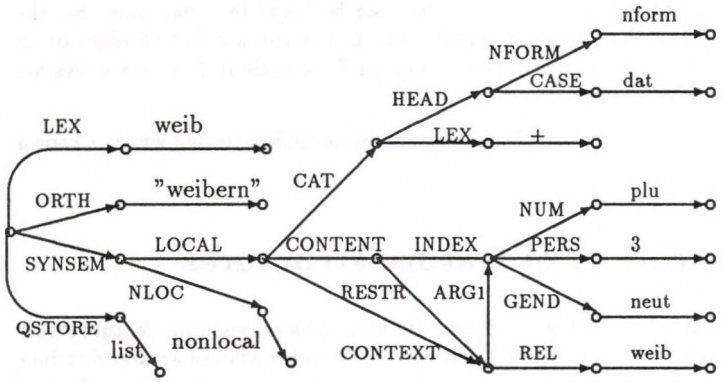
$$\left[\begin{array}{ll} \text{SYNSEM|LOCAL} & : \left[\begin{array}{ll} \text{CAT} & : \left[\text{HEAD|NFORM} : \textit{nform}\right] \\ \text{CONTENT} & : \left[\begin{array}{ll} \text{INDEX|NUM} & : \textit{plu} \\ \text{RESTR|REL} & : \textit{weib} \end{array}\right] \end{array}\right] \end{array}\right]$$

A feature structure describes some (linguistic) object by specifying values of various attributes. The value may itself be specified recursively by another feature structure or by an atomic value. We notate feature structures as *attribute value matrices (AVM)* [PS87].

Another way to interpret feature structures is to think of them as constraining the set of (linguistic) objects of the world (the lexicon) they describe. This view is appropriate for the search pattern.

The aim of a lexicon query is to complete the information given by the search feature structure with information from one or more lexical entries. The more constrained the search feature structure is, the less lexicon entries the query returns.

We also can represent feature structures as graphs:



2 The search problem

To search in the lexicon means to test each lexicon entry if it is *subsumed* by the feature structure that describes the search pattern. Since there is no unique node naming between the graph representations of the search pattern and the lexicon entry - they both have different domains - subsumption is not equal to subgraph testing.

Given two feature graphs, say S for a search pattern and L for a lexicon entry, the following procedure defines the *subsumption test:*

procedure **subsume**(S, L):
begin

(i) if S is marked with L, return **TRUE**[1] else mark S with L[2]

(ii) delete the roots of S and L, and isolate the rooted subgraphs $S_1, \ldots, S_p$ and $L_1, \ldots, L_q$.

(iii) find a homomorphism h from the rooted subgraphs of S to the rooted subgraphs of L wrt. edge labelling. If there is no homomorphism, return **FALSE.**

(iv) performe **subsume**$(S_i, h(S_i))$ for all rooted subgraphs of S.

(v) if all subsume tests of the previous step were successful then return **TRUE**, otherwise return **FALSE**

end

Two properties of feature graphs make this procedure possible:

- every feature graph has a *root node* and
- if there are two edges k and s going from node r which share the same attribute as their label, then k and s are identical.

[1]This rule eliminates cycles.

[2]If we think of numbered nodes, to mark S with L means to mark the root node of S with the number of the root node of L.

If the attribute labels are ordered, the homomorphism can be found in linear time. So, the **subsumption** procedure has $O(n)$ time complexity where n is the number of edges of an *acyclic* S. If S contains cycles, the needed time depends on L. A cycle in S with s nodes fits a cycle in L with l nodes after $\mathbf{sl}$ steps.

Subsumption can be seen as a special case of the subgraph isomorphism testing which is known to be NP-complete [Leu90].

3 Efficient encoding of feature structures

To make the search tractable it is necessary to restrict the graphs representing feature structures. Firstly, we want to focus the search on a few attributes which achieve greater discriminatory effect. Secondly, in most cases it turns out to be sufficient for the search if we think of feature structures as being trees with a fixed depth.

Next, we define appropriateness conditions for each attribute, so we know which edges can emanate from a certain node. As an example, if the labels a, b, and c are appropriate for a given attribute l^{β} then only edges labelled with a, b, or c can emanate nodes reached by l^{β}.

For the computation of complexity only binary trees of depth d are used. The lexicon has n_l entries.

3.1 Full trees

With the help of these appropriateness definitions, a space-consuming encoding of feature structures can be built as follows:

(i) construct a full tree recursively: take a root node r, for all appropriate labels *label* create edges $(r, i, label)$, take i as the new root node and proceed recursively;

(ii) introduce a boolean flag as the new edge label;

(iii) for any given feature structure copy this full tree and mark an attribute as existing in this feature structure using the boolean flag.

To give an example, for the set of labels $L = \{a, b, c\}$ and the appropriateness conditions

$$\begin{aligned} app(a) &= b, c \\ app(b) &= a, c \\ app(c) &= a, b \end{aligned}$$

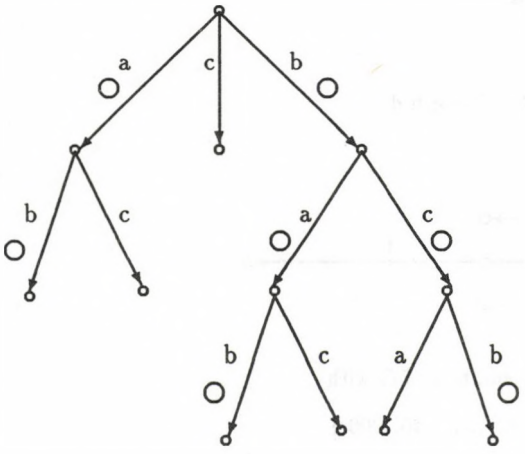
the feature structure

$$\begin{bmatrix} a : b \\ b : \begin{bmatrix} a : b \\ c : b \end{bmatrix} \end{bmatrix}$$

[3] l is a label as well.

[4] edges labeled with l

is described by the following *full tree* where the circles mark the edges which are labelled with attributes existing in our feature structure:



The subsumption procedure over full trees works as follows:

procedure **subsume**(S, L):
begin

(i) delete the roots of the search pattern S and the lexicon entry L

(ii) compare the label array of both roots. If there is at least one flag which is set in the array of S and unset in the array of L, return **FALSE**

(iii) for all activated edges from S perform this procedure recursively

(iv) if all procedure calls from the previous step succeed, return **TRUE**, otherwise return **FALSE**.

end

This procedure solves the problem in $O((2^{d+1} - 2) * n_l)$ steps.

3.2 Path enumeration

Another interesting encoding is the separation of a tree representing a feature structure into all its paths. As an example, if there are 9 possible labels extended with a zero label[5], can be assigned each path a number of base 10:

[5]The zero label indicates that between two nodes there is no labeled edge defined. *Zero* is used as a filler.

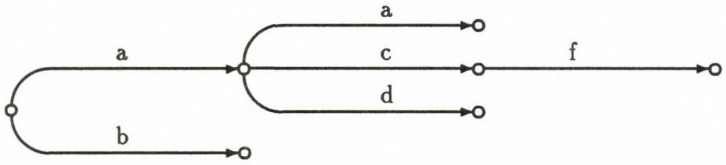
Let

(i) $L = \{a, b, \ldots, i\}$ be the set of Labels,

(ii) $h_0(l) = \begin{cases} [1..9] & : \ l \in [a..i] \\ 0 & : \ l = o \end{cases}$ and

(iii) $h_d()$ the enumeration function for paths of length d.

A possible graph over L is G



P = { aao, ado, acf, boo } is the path representation of G with

$$h_3(P) = \{110, 136, 140, 200\}$$

With the presented enumeration we get a total ordered set of all paths of all feature structures over the lexicon. Now, lexicon search can be defined in two steps:

(i) for each path of the search pattern:

- compute its path number,
- find the first element of the lexicon path set[6] which is greater than or equal to the number representing the search path, but less than the sum of the search path number and $base^x$, where x is the position of the last non-zero figure in the search path number,
- find the last element of the lexicon path set which is less than the addition of $base^x$ to the search path number.

The result of finding these two bounds is an interval, possibly empty, in which the search path subsumes all lexicon paths.

(ii) build the intersection over all the intervals.

Given a search path $p_s = a$, then for our example

(i) $h_3(p_s) = 100, \ x = 2$

(ii) $h_3(p_s) + 10^2 = 200$

(iii) the search interval is [100, 200) and

(iv) the resulting subgraph $h_3(P_{p_s})$ *is* {110, 136, 140}.

Since a full binary tree of depth d has 2^d different paths, the search of intervals for all paths is computed in maximal $2 * log_2(2^d * n_l) * 2^d$ comparisons. After 2^d subtractions the smallest interval is found and after $2^d * |Int_{min}|$ tests the intersection of all intervals results.

[6]The lexicon path set is a set of numbers

In the worst case, computing the intersection takes linear time over the lexicon size. Then the *full tree encoding* leads to better results. But in the average case, the resulting set of the search has cardinality, say, 1...10, and among the search paths there are some very discriminating ones.

With the assumption that in the average case

$$2^{d+1} \ll n_l \text{ and } Int_{min} \ll n_l$$

logarithmic time results for the lexicon search with *path encoding*, since the paths are totally ordered.

The *path encoding* yields further advantages:

(i) The number of possible paths $|P|$ can be sized down for large lexicons to $|P| < n_l$.

(ii) The search is divided into two steps. This, if wanted, gives the chance to interrupt the search after creating the intervals and to stay within logarithmic time if the minimal interval Int_{min} is too large.

(iii) After finding one interval, a decision can be taken on whether to continue the search over the whole lexicon or to create a *new ordered path set* over the lexicon elements marked with the interval.

4 On parallel searching

4.1 Hardware requirements

In order to design a parallel architecture for the search and to look for maximum speedup we want to make use of the *paracomputer model with Concurrent Read and Exclusive Write (CREW)* [Sni89].

A paracomputer consists of many identical autonomous processors, each with its own local memory and its own program. In addition, the machine has a shared memory. Each processor can simultaneously in one step read any cell in shared memory and only one processor can exclusively write to a particular cell of shared memory.

The input is given in special input cells and the output in output cells.

4.2 Maximal speedup

To achieve maximal speedup we use the *full tree encoding* of feature structures. With this encoding, a subsumption test can be performed with maximally *O(1)* where we distribute the computation over $2^{d+1} - 2$ processors with d the fixed tree depth if we regard binary trees.

The local memory of each processor consists of:

- the label l of the edge, associated to the processor,
- the relative index of this edge in the full tree,

- the number of a unique output cell which is the same for all processors of one tree and which is initialized with **TRUE**,
- the program for read, comparison, and write

The parallel subsumption can be defined as:

procedure **parallel subsume:**
for i in [1, $2^{k+1} - 2$] parallel do
begin

(i) read input cell i[7]

(ii) if i contains **TRUE** and l contains **FALSE**, try to write **FALSE** to the output cell. If write access is denied, do nothing[8].

end

The possible speedup for subsumption with the full tree encoding is linear.

For n_l lexical entries a maximal performance of *O(1)* is reachable with $n_l * (2^{k+1} - 2)$ processors and concurrent read to shared memory. The output is given in n_l output cells.

4.3 Speedup with path enumeration

The search using path enumeration is divided into the search of 2^d intervals and the computation of the resulting intersection.

The speedup for computing the intersection is linear, while the speedup for search is logarithmic. Finding a boundary takes $O(log_{p+1} n_l)$ steps. This was shown by M. Snir[Sni89]. So, O(1) is reached with n_l processors.

Note that the output computed by search with path enumeration can be more compact in comparison to parallel search with full trees.

5 A practical approach

In real applications where different modules use the resource lexicon, the problem is to maximize throughput. Standard parallel architectures like pipeline architecture are designed to maximize throughput.

In chapter 3.2 we introduced the path enumeration. This encoding gives not only an order but divides the search problem into independent subproblems. In data base systems large bases are kept tractable by defining indexes. To apply this method for lexicon search, we will build a set of index paths with the following properties:

(i) thinking of a path as a way through a rooted graph up to node n_d of depth d, n_d has to be a root of a subgraph s which for all lexical entries exists.

[7] i contains **TRUE** if this edge is specified in the input tree, otherwise **FALSE**
[8] In this case, another processor does the work

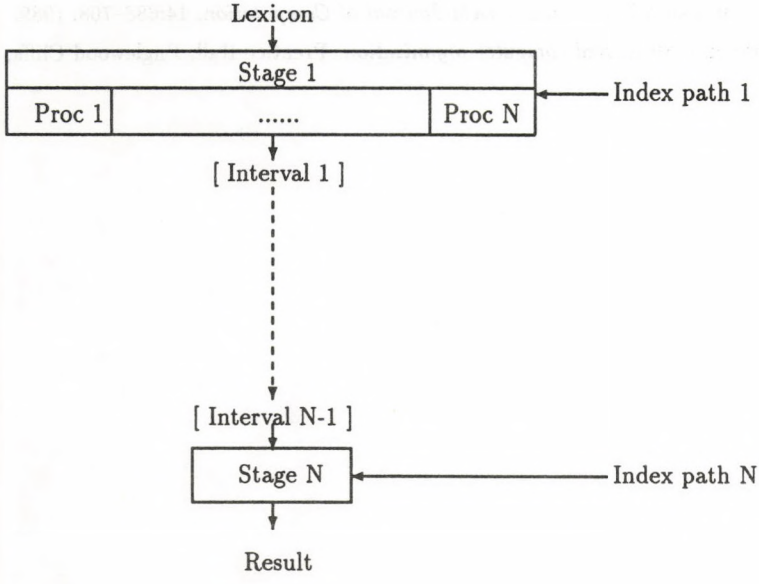
(ii) either s is atomic[9], or it is simple to define a total order over all possible s. For a given s there has to be exactly one interval $[s_0, s_1]$ that describes the possible candidates for subsumption.

(iii) the index path leads to a discriminating point w.r.t. subsumption for all lexical entries. n_d is called a decision sensitive point if s strongly selects possible solutions from the set of lexical entries.

From the set of indexes we assign one index to each stage of a pipeline.

To answer a lexicon request, the search pattern is itself searched for index paths. This results in a dynamically linked pipeline. Each stage selects a subset from its input which consists of possible subsumption candidates w.r.t. to one index path. The following stage takes this subset as input. The output of the last stage represents the resulting set of lexicon entries which the search pattern subsumes. There are as many stages in the pipeline as index paths were found.

So, the search domain gradually decreases. After finishing the selection at the first stage, the next request can start with index search while the previous request is computed in the other stages. To meet the demand of equally distributed process loads, all elements of the pipeline are themselves parallelised. Due to the decreased computational demands at later stages, the first elements of the pipeline need a higher degree of parallelisation. A possible symmetric pipeline architecture with $\frac{n}{2}(n + 1)$ processors is:



Linguistic and statistical information is needed for successful pipeline design. First we make use of linguistic knowledge to define a proper index system. To equally distribute process

[9] consists of only one edge

loads, statistical experiences and a possibly dynamic assignment from processors to pipeline stages are needed.

6 Conclusions

Lexical search is, with restrictions to the data model, simple to divide into subproblems. Under certain conditions, a space intensive encoding realizes the search problem in logarithmic time. Massive parallel systems can compute a lexicon search in one step. It is shown that parallel design of NLP algorithms can solve certain problems of time complexity in NLP applications.

References

[Kog81] P. Kogge. *The Architecture of Pipelined Computers.* Hemisphere Publishing Corporation, New York, 1981.

[Leu90] J. van Leuwen. *Handbook of Theoretical Computer Science - Algorithms and Complexity.* MIT-Press, Cambridge, 1990.

[PS87] C. Pollard and I. Sag. *Information-based syntax and semantics.* Number 13 in CSLI Lecture notes. CSLI, Stanford, 1987.

[Sni89] Marc Snir. On parallel searching. *SIAM Journal of Computation*, 14:688–708, 1989.

[Tan90] A. Tannenbaum. *Structured computer organization.* Prentice Hall, Englewood Cliffs, 1990.

The Compilation of Large Pronunciation Lexica: the Elicitation of Letter-to-Sound Patterns through Analogy-Based Networks

STEFANO FEDERICI - VITO PIRRELLI

Abstract

In this paper, we describe an analogy-based approach to automatic letter-to-sound transcription. In particular, we delve into the *analogy-based* phase of automatic elicitation of accentual patterns from a manually processed *golden set* of training data, and give the reader a glimpse of how the approach is extended to deal with the transcription process as such. The approach exhibits some advantages over other similar architectures, because of its flexibility and adaptability to different input requirements and language specificity. Its performances are illustrated and discussed. Its success shows the inherent validity of an analogy-based approach to NLP, and its capacity of modelling , with comparative ease, both core and peripheral areas of language.

Introduction.

One of the objectives of the EC-funded Onomastica project is the compilation of large pronunciation lexica of family names, first names, place-names, acronyms etc. Pronunciation lexica are an essential background resource for speech synthesis/recognition systems aimed at practical applications (Onomastica Technical Annnex, Edinburgh 1993). In Onomastica, a pronunciation lexicon consists of pairs of orthographic form/phonological transcription, such as <*Sciascia*/[ʃ'aʃ:a]>[1], possibly enriched with further information concerning the name bearer, the place of origin of the name etc. In compiling an Italian pronunciation lexicon of family names, each such pair was initially manually crafted and checked by a trained human transcriber. The sheer size of the lexicon aimed at (containing up to 1,000,000 entries), however, has soon called for a more and more substantial automatization of the transcription process. In what follows, we will describe the two-stage approach to automatic transcription adopted in Pisa. Firstly, an orthographic form is assigned a stress marker; secondly, a set of letter-to-sound rules yields the final output. In particular, we will delve into the *analogy-based* phase of automatic elicitation of accentual patterns from a manually processed *golden set* of training data. Finally, we will show how an incremental, analogy-based strategy can be used to model the transcription process as a whole.

I. Input requirements.

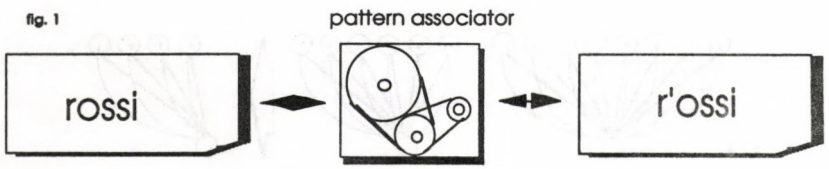
Work in Onomastica convinced us that the phonological transcription of Italian proper names can profitably be factored out into two steps: 1) individuation of the relevant accentual patterns, 2) individuation of letter-to-sound correspondences.

This has been felt convenient for a number of both practical and theoretical reasons (Pirrelli and Salza 1993, Church 1986, Salza 1990). Step number one does not lend itself to a formalization in terms of yes-or-no rules, due to the inherent difficulty of stating whether an Italian accentual pattern is regular or exceptional: in many cases, the same Italian family name can show more than one accentual pattern (e.g., b'arbaro/barb'aro, r'iccioli/ricci'oli etc.), depending on the place of origin of the name, or even on the personal taste of the name owner. Stress assignment is then carried out by a self-learning software (the **STRESS** system) that a) does not make any principled distinction between rules and exceptions (a rule is only a cluster of often-recurring "exceptions"), b) formulates conditions on the application of accentual patterns in terms of a continuum (the analogy of unknown items to already known ones), rather than in terms of mutually exclusive classes of phenomena, c) continuously and automatically updates its linguistic knowledge whenever its analysis fails. The training sample is a

golden set of 25,000 manually stressed proper names in orthographic form. The training list looks like the following:

r'ossi, ferr'ari, esp'osito, r'usso, bi'anchi, col'ombo, etc....

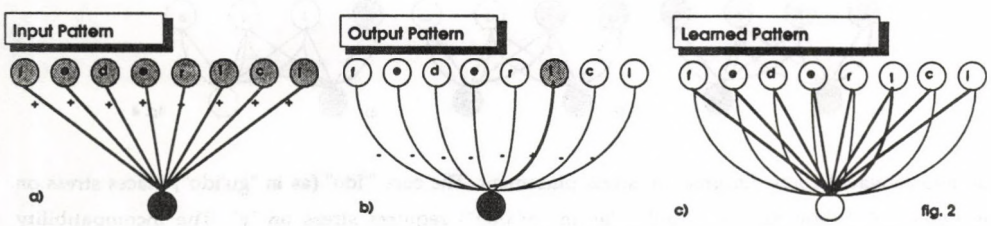
STRESS can be seen as a pattern associator which takes as input a raw string of characters and outputs the string plus stress (fig.1):



Training consists in exposing the system to input/output pairs, so that STRESS will gradually find out what sorts of regularities underpin the string/stress associations, and will be able to replicate them.

II. The learning routine.

The input string is seen by STRESS as a *pattern of nodes* in its network. As far as STRESS is concerned, all characters are, initially, potential stress carriers. In fig.2a, this is indicated by the fact that all nodes are positively activated (grey circles).



In the output pattern (fig.2b), only one (or more) vowel(s) remains activated, namely the vowel(s) where stress is placed. All other characters are turned off (white circles). On the basis of these two patterns, STRESS builds up a so-called *learned pattern* (fig.2c). Information about the stressed vowel is conveyed by an activation link (*+links*, darkened in the diagram). All other links are inhibitory links (*-links*, indicated as lighter links in the same diagram). The sequence of characters in fig.2c defines the

context where the pattern applies. In fact, STRESS is capable of extracting smaller patterns than the one in fig.2c, on the basis of the analogies between already learned patterns. This is illustrated by fig.3, where two learned patterns show something in common (namely the ending "r'ici") and something different. STRESS will extract the common core, and will use it as an independent learned pattern.

Preliminary conditions for extraction are: i) the two learned patterns have one or more links in common (that is, links to the same characters, in the same position in the string), and ii) links' signals match (*e.g.*, a *+link* does not match with a *-link* etc.). Otherwise, core extraction does not take place.

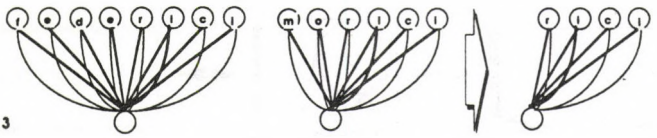
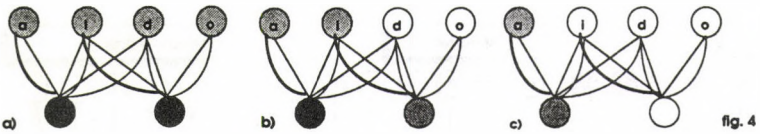


fig. 3

It should be borne in mind that these extraction principles have not been tailored to the speciifc task at hand: in fact, they have independently been devised for general purpose language applications (Federici 1990), and successfully tested on a wide spectrum of NLP problems pertaining to different domains of linguistic analysis.

How does STRESS conjure up a particular response? Let us consider a concrete case. In fig.4, two previously learned cores are simultaneously activated by the same input string, the string "braido" (whose correct pronunciation is with stress on "a": "br'aido").



a) b) c) fig. 4

The two activated cores disagree on stress placement. The core "'ido" (as in "gu'ido") places stress on the vowel "i", while the core "'aid" (as in "m'aida") requires stress on "a". The incompatibility between the two patterns is expressed by their sending each other inhibitory signals (the lighter arcs). Step a) in fig.4 pictures a stalemate situation, where neither patter wins out over the other, since they have the same total number of supporting/inhibiting units. The network starts cycling. In the cycling, the interplay of inhibition/activation links turns off two nodes: "d" and "o" (as shown in the diagram b of fig.4). This causes both activated cores to lose some of their supporting units down the road. The reader will note that the activation loss of the core "'ido" is bigger than the one of "'aid", the latter

being still supported by two units (namely "i" and "a"), while "aid" remains with one supporting unit only. At the end of the cycle (diagram c in fig.4), "'aid" will then win out over "'ido", as the stalemate is eventually broken. This has an intuitive justification: the pattern "'aid" says something about two potentially stressable vowels, both "a" and "i", and gives preference to one of them only; on the contrary, "'ido" is less informative in the context at hand, since it ignores the existence of a stressable "a" to the left of its candidate stressed vowel.

In some cases, however, the stalemate cannot be broken, since the very same pattern supports/inhibits two (or more) rival accentual patterns (as in the pair b'arbaro/barb'aro). A real ambiguity is thus engendered, since all rival nodes remain activated. As a result, STRESS will output all possible alternative stress placements.

III. Performances and Improvements.

STRESS' performances were tested on a number of lengthy runs. Results were extremely encouraging: success rates are shown in the table below, where different ways of coding the input string (what we have called *chunking strategies*) are tested against training data of growing size (*training samples*). A chunking strategy implies the definition of the relevant primes in input, that is what unit constitutes a single node in the input layer. During testing, we have been entertaining the following three hypothesis (in this succession): 1) that primes are individual characters; 2) that primes are syllables (or, to be more precise, an approximation of the established notion of syllable); 3) that primes are obtained by extracting a vowel skeleton and a consonant skeleton from the input string. Strategy number 3 is actually a kind of hybrid of both 1 and 2. Figure 5 illustrates the result of adopting the three chunking strategies at the level of input coding, when the string "betti" is fed into STRESS. The table of performances overleaf proves that the last routine turns out to be more effective. Theoretically, the result comes to no surprise, and is given indirect confirmation by the fact that consonantal length is known to play a prominent role in determining stress placement in Italian as well as in other languages (as also pointed out in Church 1986).

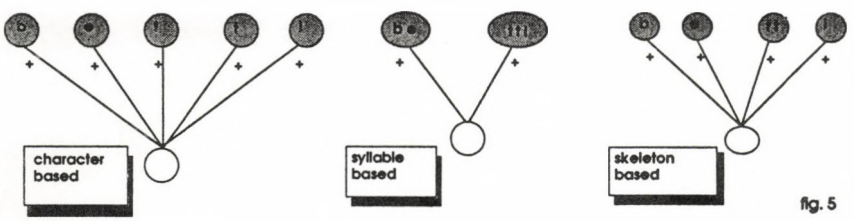


fig. 5

The strategy illustrated so far presupposes that letter-to-sound transcription in Italian is thoroughly rule-governed. However, some classical "thorny" clusters for Italian letter-to-sound rules, such as "i" followed by a stressed vowel, are even more difficult to tackle through rules when it comes to proper names, as witnessed by false friends such as *Ghiotto* [gj'ot:o] and *Ghione* [gi'one]. Moreover, dialectal idiosyncracies make the mapping between spelling and phonetic transcription less predictable. The proper name *Maxia* (from Sardinia), normally transcribed as [m'aksja], is actually pronounced [maʃ:'ia] and thus conversely mispelled as *Mascia* in those places where the name sounds new or unfamiliar.

strategy/training performances

chunking strategy \ training sample	2000	10000	20000
character	87%	90%	91%
syllable	29%	57%	66%
skeleton	84%	91%	93%

It is clear that only the use of comprehensive computerized repertoires of exceptions, whose intended purpose is to curtail the domain of more regular processes through full listing of relevant exceptional contexts, can improve on the performance of a rule-based transcriber.

In (Federici, Pirrelli 1992; Federici, Pirrelli 1993) we showed that an analogy-based network can equal the performances of a rule-based morphological parser. Accordingly, the transcription process as a whole can be modelled by an analogy-based mechanism, through the architecture sketchily illustrated in fig.6, where an extra output layer has been introduced in order to account for the level of transcription proper.

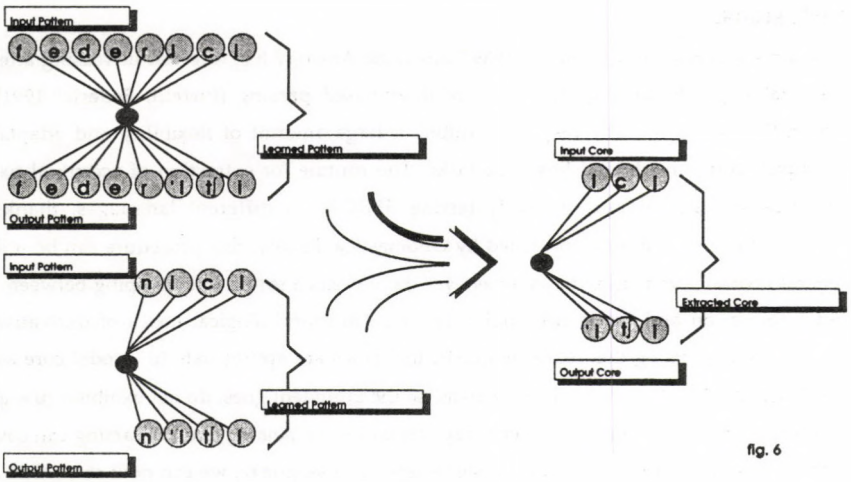


fig. 6

In fig.6, the transcription of the string "ici" as /'itʃi/ is the result of the occurrence of the same string in two different names whose endings are pronounced according to the same sequence of sounds. The general architecture is reminiscent of the approach taken for the program PRONOUNCE (Dedina, Nusbaum 1986), although our analogy-based routine departs from the analogy-based routine implemented in PRONOUNCE in that not every possible match at the string level is used as an indicator of a possible analogy at the level of sound (as done by Dedina and Nusbaum): only those matches which cluster into relevant pronunciation patterns are subsequently adopted in attempting to pronounce unknown strings. STRESS learns not only what units should be looked at, but also how many of them should be looked at together. In the literature, the number of letters which are treated as the left and right context of the target letter to be pronounced is called a window. Usually, in neural network architectures for letter-to-sound correspondences (such as NETtalk, Sejnowsky, Rosenberg 1986) a window is defined at the outset once and for all. This creates well-known problems of mispronunciation due to lack of context, as in the English pair *nation/national* with a seven letter window. STRESS avoids these problems, since it adopts a different window span depending on the input string it has to pronounce. In other words, STRESS learns also how to "window" a word.

Finally, our analogy-based routine is based on a set of principles of analogy whose generality has been already successfully tested in a variety of applications.

V. Conclusions.

Since Skousen's seminal work (Skousen 1989), linguistic Analogy has received increasing attention by computational linguists. Our approach to analogy-based parsing (Pirrelli, Federici 1991), which differs from Skousen's in many respects, exhibits a large amount of flexibility and adaptability to different input requirements and linguistic tasks. The routine for extraction of accentual patterns is language independent. We are currently testing STRESS on different languages, thanks to the multilingual exchange framework provided by Onomastica. Finally, this procedure can be utilized for more general lexicon compilation tasks, whenever there exists a systematic mapping between (part of) the input information and the lexical coding (as, e.g., in morphological lexica of derivatives). It is commonly assumed, among theoretical linguists, that rules are appropriate to model core aspects of language (Chomsky 1981). Some peripheral areas, so the argument goes, do not exhibit a rule-governed behaviour and lend themselves to a rather fuzzy formalization. Analogy-based parsing can cover these areas with comparative ease. However, for some aspects of language, we can now show that Analogy explains also non-negligible portions of the language core. It would not be surprising if these findings will lead to a change of perspective in the way of thinking about the role of linguistic rules *tout court*.

References.

Chomsky N. , 1981 Markedness and Core Grammar. in Belletti, Brandi, Rizzi (eds.) *Theory of Markedness in Generative Grammar.* Proc of the 1979 GLOW Conference, Pisa, pp.123-46.

Church, K. 1986 Stress assignment in letter to sound rules for speech synthesis, Proc ICASP, vol.4 Tokio.

Dedina M.J., H.C. Nusbaum 1986, PRONOUNCE: a program for pronunciation by analogy. in Proceedings of the 14th Annual ACM Computer Science Conference, Cincinnati.

Federici S., 1990, Un sistema connessionista autoespandibile di comprensione del linguaggio naturale. Laurea Dissertation in Computer Science, Pisa University.

Federici S., V. PIrrelli 1992 A bootstrapping strategy for lemmatization: learning through examples. Proceedings of Complex.

Federici S., V. Pirrelli 1993 The representation of combinatorial structure in parallel networks: ANTAEUS a self-modelling tagger. ICML Proceedings, Barcellona.

Onomastica Technical Annnex, Part III, Research Project Proposal , Edinburgh 1993.

Pirrelli V., S. Federici 1991 An analogical way to language modelling: MORPHEUS. Acta Linguistica Hungarica (41).

Pirrelli,V., P.L. Salza 1993 Transcription rules and stress assignment in Onomastica. ILC/CSELT-R-0001.

Salza, P.L. 1990 Phonetic Transcription Rules for Text-to-Speech Synthesis of Italian. Phonetica, 47.

Sejnowsky T.J., C. R. Rosenberg, 1986, NETtalk: a parallel network that learns to read aloud. *The Johns Hopkins University Electrical Engineering and Computer Science Technical Report JHU/ECS-86/01.*

Skousen, R.1989 *Analogical Modeling of Language* Dordrecht, Kluwer.

---

[1] Hereafter, stress will be indicated in pre-vocalic position.

Lexical Access in an Integrated Speech-Language System

GUNTER GEBHARDI

Abstract

The paper gives a sketch of an advanced architecture for a speech–language recognition and comprehension system and looks at the special demands of the lexicon in the system. In particular, this paper discusses aspects of lexical access. The result is useful for lexical access with random keys and underspecified lexical entries in unification based systems.

1 Introduction

1.1 Motivation

Currently, a lot of research projects are going in the field of speech–language systems. Most of the systems are *word oriented* – regarding the problem of speech comprehension as a problem of recognition of a sequence of words. So the most important task to solve is the mapping of the input signal onto a word. The first objection to this approach is: Usually it is possible to map an input signal to a *set* of words, only. The second objection to the approach: Speech is not restricted to being a sequence of words, speech is a sequence of particles, words, phrases, groups of words, ...

1.2 The Architecture

One aim of the ASL–Project[1] is to design an architecture ([HAHN92], [PYKA92], ideas of [BRISCOE87]) of a speech recognition and comprehension system which

[1] ASL – Architectures for Speech and Language recognition systems.

is not specifed for this crucial mapping function. The system should handle incomplete input information (information which cannot be mapped on *one* word) in an efficient way and should not be specifed for the recognition of words. One basic idea comes from cognitive science. The idea is to include expectation information. Understanding an utterance is the unification of the interpretation of speech events and expectation attitudes.

2 Lexical Access

Look at the lexicon as a part of the system. It should be obvious that the knowledge base consists of a word oriented part, the lexicon, and a part with descriptions of particles, phrases, groups of words[2]... (What kind of grammar is able to handle such phenomena?)

The lexicon is built up on feature structures. For the sake of convenience the structures here are restricted to those of [SHIEBER86]. Each feature and set of features can serve as a key for access. Here a problem arises, concerning the nature of feature structures and the way to use them.

2.1 The idea and the problem of underspecification

Underspecification is a very elegant and efficient method for lexical description.

A very simple lexicon example giving the inflexion of German nouns is used to explain the ideas. The two structures

$$(1)\quad \begin{bmatrix} ORTH : < Blume > \\ AGR \begin{bmatrix} NUM : sing \\ GEN : fem \\ CASE : nom \end{bmatrix} \end{bmatrix} \qquad \% \; flower,\ nom$$

$$(2)\quad \begin{bmatrix} ORTH : < Blume > \\ AGR \begin{bmatrix} NUM : sing \\ GEN : fem \\ CASE : gen \end{bmatrix} \end{bmatrix} \qquad \% \; flower,\ gen$$

are entries for the lexeme *Blume* of the lexicon. (1) gives the information singular, feminin, nominative and (2) gives the information singular, feminin and genitive. It is simple to imagine the entries between the accusative and the dative form: the only difference of all those entries is the value of the case information. If it is impossible to determine or to exclude a property encoded in feature structures, the feature structure can be left *underspecified* with respect to this information. Because it is impossible to determine the case of *Blume*, the resulting lexicon entry can be left as

[2] anything between two breaks in an utterance

(3) $\left[\begin{array}{l} ORTH : < Blume > \\ AGR \left[\begin{array}{l} NUM : sing \\ GEN : fem \end{array}\right] \end{array}\right]$ % *flower*

To explain the problem and the solution the lexicon will be expanded with some additional entries:

(4) $\left[\begin{array}{l} ORTH : < Auto > \\ AGR \left[\begin{array}{l} NUM : sing \\ CASE : nom \vee dat \vee acc \\ GEN : neut \end{array}\right] \end{array}\right]$ % *car*

(5) $\left[\begin{array}{l} ORTH : < Teller > \\ AGR \left[\begin{array}{l} CASE : nom \vee acc \\ GEN : masc \end{array}\right] \end{array}\right]$ % *plate*

(6) $\left[\begin{array}{l} ORTH : < Gedanke > \\ AGR \left[\begin{array}{l} NUM : sing \\ CASE : nom \\ GEN : masc \end{array}\right] \end{array}\right]$ % *thought*

The lexicon entry of (5) is also underspecified – with respect to the number value.

By using underspecification the number of entries in the lexicon and to be handled during processing can be restricted. As a result the systems will run more efficiently.

Usually, the lexicon access operation is restricted to a simple table look up (with a simple or a complex key). Fortunately, this operation can be implemented very efficiently.

Doing that the basic assumption is that there is (at least) one never underspecified feature with a value giving a identification to the entry. The *ORTH* value may have these properties. To solve the query

(7) $\left[ORTH : < Gedanke > \right]$

the value of the *ORTH* feature is taken and gives access to (6).

More generally, but mostly theoretically, the lexical access operation is subsumption. The result of lexical access is the set of entries M with

(8) $M = \{m | m \in LEXICON \wedge QUERY \sqsubseteq m\}$

Now consider the query

(9) $\left[AGR \left[CASE : acc \right] \right]$

This query may be (part of) the hypotheses (subcategorization information) for the unknown part *X* of a sentence like *Er sah X*.

It should be obvious that the result contains information of (4) and (5). But the resulting set also has to contain information of (3)! Although there is no information about case in (3) – which is underspecified with respect to case information – (9) unify (3):

(10)
$$\begin{bmatrix} ORTH : < Blume > \\ AGR \begin{bmatrix} NUM : sing \\ GEN : fem \\ CASE : acc \end{bmatrix} \end{bmatrix} \qquad \% \; flower$$

The function of lexcial access has to change. The assumption that there is one never underspecified feature with a value giving an identification to the entry does not hold. The condition to find the set of entries M is in general

(11) $M = \{m | m \in LEXICON \wedge QUERY \sqcup m\}$

or restricted to subsumption

(12) $M = \{m | m \in LEXICON \wedge \exists n(m \sqsubseteq n \wedge QUERY \sqsubseteq n)\}$

So, if underspecification is permitted and the access key not strongly restricted, the function for the lexical access in the system becomes more complicated. The access operation has to be unification! But unification is a very expensive operation. This is the resulting problem of underspecification here!

3 The Access Problem Revisited

3.1 The Problem of Different Keys

The usual means of lexical access is to use the lexeme as the key. In language generation systems a well defined subset of the lexical description serves as a complex key.

Here the system properties make it impossible to use the lexeme or such a subset as the key for lexical access. A query for a lexical object consists of one or a number of properties (features) of a lexical object (which are possibly underspecified in the lexicon). In speech recognition systems it depends on the quality of the signal and of the signal recognition, on syntactic or semantic properties. So, it is impossible to define a key. To use each feature as a key is no solution, because sometimes such keys address half of the lexicon. To calculate all possible key combinations is always no solution, because the number becomes very large. How to avoid *linear* growing of search space without keys?

3.2 The Costs of Unification

Unification is a very expensive operation. It is too expensive to check wether an entry fits the query or not. To take an entry without unification of the entry and the query is impossible because of the (possible) loss of information. The task is to find out a way something inbetween.

4 The Approach for Solution

4.1 The Principle

To improve the search, a lot of features or sets of features serve as *subkeys*. Each subkey determines (addresses) a set of lexicon entries. The decision is made by the unification test – before run time!

A query to the lexion will be divided into a set of subkeys. Step by step, the query will be reconstructed by the unification of the subkeys. Simultaneously, the intersection of the addressed lexical entries will be calculated. The unification of two subkeys reflects the intersection of the addressed lexicon entries. The result is the restricted set of lexical entries.

If not all features are subkeys, each member of the set of resulting lexical entries has to be unified with the query. But this set is small.

In this way two solutions for the problems of lexical access under the special system demands can be obtained. Firstly: It is possible to avoid or to restrict the very expensive unification of large lexical entries with the query on run time. Unification is mostly restricted to the small subkey structures. Secondly: The access to a large number of lexical entries will be delayed and is restricted to only a small number. For random queries the search effort is equivalent to the *log* of the number of entries.

4.2 Discussion and Example

The basic idea is to restrict unification during *on–line* processing and to extent the indexing technique *off–line* to include unification.

4.2.1 Off–Line Operation

The technique of additional data structures, called indexes, to get a speed up for databases access is well known [ELMASRI/NAVATHE89], [ULLMAN88].

The primary concept to calculate the index here is the usage of *subkeys*. Informally, a subkey is defined as a feature structure which has to subsume on the one hand side a query and on the other hand side a *subset* L' of entries of the lexicon L with $|L'| < |L|$. In difference to the definition of keys in database technology a subkey needs not to be part of an entry.

As an example the following subkeys are defined (for discussion see 5.1):

(13) $\left[\, AGR\left[\, CASE : nom \,\right] \right]$

(14) $\left[\, AGR\left[\, CASE : acc \,\right] \right]$

(15) $\left[\, AGR\left[\, GEN : masc \,\right] \right]$

(16) $\left[\, AGR\left[\, NUM : sing \,\right] \right]$

The table

Entry ⇓ Key ⇒	(13)	(14)	(15)	(16)	...
Blume % flower (3)	$\sqcup$	$\sqcup$		$\sqsupseteq$	
Auto % car (4)	$\sqsupseteq$	$\sqsupseteq$		$\sqsupseteq$	
Teller % plate (5)	$\sqsupseteq$	$\sqsupseteq$	$\sqsupseteq$	$\sqcup$	
Gedanke % thought (6)	$\sqsupseteq$		$\sqsupseteq$	$\sqsupseteq$	

shows which subkey subsumes or unifies which lexicon entry.

4.2.2 On–Line Operation

The first step to get on–line access to the lexicon is to select all subkeys which subsume the query. The selection of possible subkeys is guided by an analysis of the query. The set of possible subkeys can be restricted and not each subkey has to be used for subsumption test. For example the query

(17) $\left[\, AGR\left[\begin{array}{l} NUM : sing \\ CASE : acc \\ GEN : masc \end{array}\right] \right]$ % *query*

contains no information about case nominative, therefore the test (17) $\sqsupseteq$ (13) can be left out.

The sequence of checking the subsumption relation is determined by the size of the addressed sets of lexical entries. The aim of this ordering is to restrict the set of possible lexical entries as early as possible to get more efficiency.

The order of subkeys subsuming the query is (15), (14) and (16).

In parallel the intersection of the addressed lexical entries has to be calculated. The intersection corresponds to the unification of the subkeys. The result of this unification is the *key* which is the equivalent used for lexical data base access. Because the key subsumes the query, it should be obvious that the result of the lexical access gets *at least* all possible lexical entries.

The addressed sets in the example are: $\{(5),(6)\}$ by (15), $\{(3),(4),(5)\}$ by (14) and $\{(3),(4),(5),(6)\}$ by (16). The intersection of these sets is $\{(5)\}$.

Now, the set of possible entries is restricted, the system gets as a second step the entries of the lexicon. Here it is only (5).

The last step is to unifiy the set of possible entries and the query. All those entries succesfully unified with the query satisfy *at least and at most* the query.

In the example the result is

$$(18) \quad \left[\begin{array}{ll} ORTH : < Teller > & \\ AGR & \left[\begin{array}{l} CASE : acc \\ GEN : masc \\ NUM : sing \end{array}\right] \end{array}\right]$$

5 Remarks

5.1 How to Get Subkeys?

5.1.1 The *Granularity* of Subkeys

On the one hand the set of addressed lexical entries of such simple subkeys like those shown in (13) and following is very large in a real lexicon. If so, a larger number of calculation steps are necessary to get the key for lexcial access.

On the other hand a subkey may only address one lexicon entry and the subkey becomes the key. No calculation effort is necessary, but it is more complicated to select the subkey.

This dilemma is called the question of granularity. A general solution remains to solve.

5.1.2 Definition of Subkeys

The definition of subkeys is a central question in respect to the presented method of lexical access. In general, there are three possible ways to define the subkeys:

1. Automatic Definition

   The automatic definition is founded upon an analysis of all lexicon entries. Each lexicon entry l is divided into a set of very fine subkeys k, with $l \sqsupseteq k$. All extracted keys of each lexicon are put together and serve as subkeys for the whole lexicon.

   The advantage of this principle is that it can be done automatically. The disadvantages are the mostly large number of subkeys, subkeys which will never be used and the super fine granularity of some subkeys, i.e. subkeys which will always be used in a special combination.

2. Definition by Hand

   It is possible to define the subkeys by hand without any implementation effort, by means of the knowledge about the processing strategies of the

system (e.g. the parser). It may be a solution to define the keys once, but it is crude for frequently definition. Besides the question of how to up date the subkeys, this method is a source of errors.

3. Definition by Training

   The definition of the subkeys by training or learning is mostly the best method. The queries to the lexicon will be recorded and later analysed like described in the method of automatic definition. In contrast to this method the extracted keys here are really used keys. The recording protocol may also be used to decide the granularity of the subkeys.

It is impossible to give a general desicion rule for method of definition. This depends on the application. However, it is expensive to get efficieny and flexibility.

5.2 Types

Till now the feature structures of discussion were of the kind of [SHIEBER86]. But it is also possible to store information including type information in the lexicon and to use the same access formalism.

Are there advantages using type information for lexcial access?

Consider wrong queries like

(19) $[AGR[NUM: fx42]]$

(20) $[VACUUM[CLEANER: fx42]]$

Without types and appropriateness conditions it is impossible to reject such queries. The result of the lexicon access with (20) using the usual way of lexical access (subsumption) is an empty set. But the result of the lexicon access with (20) using unification as selection condition is the whole lexicon!

Using typed feature formalisms ([CARPENTER92]) it is possible to reject such queries.

Moreover, is it possible to use type information to restrict the search space?

The first answer is that it is also impossible to make a strong type prediction with weak information and an universal type does not restrict the search space.

The second answer is a speculation. Using a type lattice including the whole lexicon and compiling this structure in a computational very efficient form, it may be that the lexicon access becomes very efficient. It is to settle the complexity of the problem. However, this method is restricted to a complete lexicon.

6 Conclusion

The aim of the paper was to give an informal overview of the designed method for lexical access. So the proofs of soundness and completeness were left out here.

A programm founded on the method described in the paper serves as a lexicon interface in a bread–board environment for experimental research with speech–recognition systems.

The architecture of an integrated modular speech–language system causes some problems for lexical access. It is possible to solve these problems and so to support the new system idea.

However, many problems remain to be solved.

References

[BRISCOE87] Briscoe, E. J. (1987): *Modelling Human Speech Comprehension: A Computational Approach.* Ellis Horwood, Chichester.

[CARPENTER92] Carpenter, B. (1992): *The Logic of Typed Feature Structures.* Cambridge University Press, Cambridge.

[ELMASRI/NAVATHE89] Elmasri, R. and Navathe, S. B. (1989): *Fundamentals of Database Systems.* Benjamin/Cummings Publishing Company, Redwood City, Cal.

[HAHN92] Hahn, W. v. (1992): *Von der Verknüpfung zur Integration: Kontrollstartegie oder kognitive Architektur.* in: Görz, G. (eds)(1992): *Konvens 92.* Springer–Verlag, Berlin.

[PYKA92] Pyka, C. (1992): *Management of Hypotheses in an Integrated Speech–Language Architecture.* Proceedings ECAI 92 (10th European Conference on Artifical Intelligence). John Wiley & Sons, Chichester.

[SHIEBER86] Schieber, S. M. (1986): *An Introduction to Unification–Based Approaches to Grammar.* CSLI Lecture Notes Number 4, Menlo Park, Cal.

[ULLMAN88] Ullman, J. D. (1988): *Principles of Database and Knowledge–Base Systems.* Computer Science Press, Rockville, Md.

What is a Word, What is a Sentence? Problems of Tokenization

GREGORY GREFENSTETTE – PASI TAPANAINEN

Abstract

Any linguistic treatment of freely occurring text must provide an answer to what is considered as a token. In artificial languages, the definition of what is considered as a token can be precisely and unambiguously defined. Natural languages, on the other hand, display such a rich variety that there are many ways to decide upon what will be considered as a unit for a computational approach to text. Here we will discuss tokenization as a problem for computational lexicography. Our discussion will cover the aspects of what is usually considered preprocessing of text in order to prepare it for some automated treatment. We present the roles of tokenization, methods of tokenizing, grammars for recognizing acronyms, abbreviations, and regular expressions such as numbers and dates. We present the problems encountered and discuss the effects of seemingly innocent choices.

1 Introduction

The linguistic exploitation of naturally occurring text can be seen as a progression of transformations of the original text. The original text is a sequence of characters. Before any syntactic analysis of the corpus is performed, two transformations usually take place. Sentences must be isolated since most grammars describe sentences. And, in order for sentences to be isolated, words must be isolated from the original stream of characters. The isolation of word-like units from a text is called tokenization. The results of this tokenization are two types of tokens: one type corresponding to units whose character structure is recognizable, such as punctuation, numbers, dates, etc.; the other type being units which will undergo a morphological analysis.

In linguistic textbooks tokenization is quickly dispatched as a relatively uninteresting preprocessing step performed before linguistic analysis is undertaken. In reality, tokenization is a non-trivial problem. Confronted with large corpora of raw text, the computational lexicographer must come to grips with the transformations presented schematically in Figure 1 and make the difficult choices, choices whose repercussions are sometimes only felt long after.

In this paper we will discuss the choices that must be made, how they can be made, when they should be made and their possible effects on subsequent linguistic treatment.

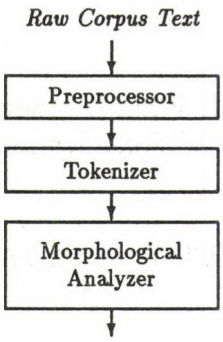


Figure 1: Text Transformations Before Linguistic Analysis.

2 Preprocessing

We will consider throughout that we are dealing with a text in electronic form as a sequence of characters, rather than a scanned image of text. Electronic text is readily available these days, in increasing numbers, usually produced as a by-product of typesetting. Such text often contains extra whitespace and a number of mark-ups that indicate font-changes, text subdivisions, special characters, and a hundred other things. Although such indications carry meaning — they are there to help the reader understand the text — they are usually filtered out from the text in a preprocessing stage before any linguistic processing, or even before tokenization begins.

Since little normalization exists in typesetting codes, we will not discuss the matter further, except to provide a method of eliminating SGML-type code from a running text[1]. Unix-based workstations furnish a general-purpose character stream scanner called *lex* or *flex*. This scanner permits the definition of actions to be taken when certain regular expressions are matched in the input text. Figure 2 provides a simple *lex* program [2] which deletes SGML markings from an

[1] A public domain SGML parser called *SGMLS* is available from the anonymous *ftp* site ifi.uio.no. (129.240.64.2) in the directory /pub/SGML/SGMLS. This parser allows much finer handling of SGML codes.

[2] The notation for the regular grammars shown here are the following:
. matches any character except newline.
^ matches the beginning of a line.
$ matches the end of a line.
\n matches the newline character.
[abc...] character class, matches any of the characters abc...
[^abc...] negated character class, matches any character except abc... and newline.
r1|r2 alternation: matches either r1 or r2.
r1r2 concatenation: matches r1, and then r2.
r+ matches one or more r's.
r* matches zero or more r's.
r? matches zero or one r's.
(r) grouping: matches r.

```
/*  Call this file StripSGML.lx, and then run:
flex -8 -CF StripSGML.lx; gcc -o StripSGML lex.yy.c -lfl -s
To pass this simple filter over a text file called toto,  run:
StripSGML < toto                                                    */
%%
"<"[^\n<>]+">"  ;
.               ECHO;
[\n]            ECHO;
%%
```

Figure 2: Flex program for filtering out SGML markings.

```
/*  Call this file dehyphen.lx, and then run:
flex -8 -CF dehyphen.lx; gcc -o dehyphen lex.yy.c -lfl -s
    To pass this simple filter over a text file called toto,  run:
dehyphen < toto                                                     */
%%
[a-z]-[ \t]*\n[ \t]*    { printf("%c",yytext[0]); }
%%
```

Figure 3: Flex program for dehyphenating a text.

input file.

Not only do some things have to be filtered out of marked-up text, some things have to rejoined. The most common case that appears in raw text is hyphenation at right margins. Since this hyphenation is usually only circumstantial, related to the width of the page and not to the meaning of the text, one might easily consider eliminating it from text files that employ it. The short *lex* program of Figure 3 eliminates a trailing hyphen from a text and rejoins the hyphenated word to its second half on the next line. The regular expression that the filter recognizes is a lower-case letter, followed by a hyphen, then any number of tabs or spaces, followed by a newline character and more spaces. Only the alphabetic character is retained and printed out by the filter. All other characters in the file pass through unchanged.

Of course, introducing hyphenation into a text during typesetting can produce lines ending in a hyphen not because the word was split there, but because a naturally occurring hyphen happened by chance to appear where the word would be split. Suppose that the word *small-town* was split at the end of line by the typesetting, then this filter would return the string *smalltown* as one token. In order to test just how often this might happen in reality, we took the Brown corpus (Francis and Kucera, 1982), a corpus whose tokenization was hand corrected, and ran it through a typesetting program (*nroff*) which introduced end-line hyphenations. The Brown corpus contains about 1 million words. Typesetting the untokenized Brown corpus produces 101860 lines of formatted text, of which 12473 (12%) ended in a letter plus hyphen. Joining these lines using the filter given in Figure 3, produced 11858 correct dehyphenations and 615

errors (4.9%), i.e. words which did not appear in the original text. Examples of erroneously joined words are *ring-aroundthe-rosie, rockcarved, rocketbombs, rockribbed, roleexperimentation, rookie-of-theyear, satincovered, sciencefiction.* This experiment gives a taste of the type of choices that must be made during tokenization. Here, if one had access to a dictionary and morphological package at this stage, one could test each of the 12473 cases by analyzing the constituent parts and making more informed decisions, but such a mechanism is already rather sophisticated, and its construction is rarely considered for such a preliminary stage of linguistic treatment. One may consider the 615 errors (out of 1 million words) as so many unknown words to be treated at some later stage, or just accept them as noise in the system.

3 Roles of Tokenization

Once the input text of the corpus is preprocessed, we have a string of characters corresponding to what the linguistic processors will consider as the text. At one stage in this linguistic processing the elements of the text will be considered as belonging to a certain syntactic class. For example, the string *dog* will be considered as a SINGULAR-NOUN. In order for classes to be assigned to strings, the original text, which can be considered as one long string, has to be divided into units which will be recognized as members of a class. One traditional role of tokenization is the recognition of these units.

The other traditional role of tokenization is the recognition of sentence boundaries, since most linguistic analyzers consider the sentence as their unit of treatment. We will consider this traditional view here, demonstrate how it can be implemented, and show its limitations in handling certain ambiguous cases of word and sentence boundaries.

4 What is a word, What is a sentence?

Sentences end with punctuation. The exclamation point and the question mark are almost always unambiguous examples of such punctuation. But the period is an extremely ambiguous punctuation mark. It is not trivial to decide when it is a full-stop, a part of an abbreviation, or both. In the Brown corpus, there are 48885 sentences and 3490 (1 in 14) contain at least one non-terminal period.

Isolating word and sentence boundaries involves resolving the use of ambiguous punctuation. The second role of tokenization is, then, the one which must be attacked first. Some structurally recognizable tokens contain ambiguous punctuation, such as numbers, alphanumeric references (e.g. *T-1-AB.1.2*), dates (e.g. *02/02/94*), acronyms (e.g. *AT&T*), punctuations, and abbreviations (e.g. *m.p.h.*). Some of these classes can be recognized via regular expression grammars which predict the structure of the tokens as will be illustrated below. Once these units are recognized the only uses of separators are non-ambiguous, and they can thus be used surely to delimit words and sentences.

4.1 Ambiguous Separators in Numbers

Numbers are the least ambiguous of the structural types. Still, the structure of numbers are language specific constructions, for example the English number *123,456.78* will be written as *123 456,78* in French newspaper text. A *lex* regular expression which recognizes the English version of numbers is **([0–9]+[,])*[0–9]([.][0–9]+)?** while a regular expression accepting

the French version is ([0–9]+[␣])*[0–9]([,][0–9]+)?. These expressions would *overgenerate* strings, outside the class of numbers, but one rarely sees strings such as *12,45.678* in ordinary text, and even if one did one would probably want it considered as a number.

[0-9]+(\/[0-9]+)+	date
([+\-])?[0-9]+()?[0-9]*%	percent
(\[0-9]+,?)?+(\.[0-9]+|[0-9]+)*	numbers, dollars

The above table gives some regular expressions for English numbers, dollar value and date-like constructions that can incorporated into a tokenizer. Recognizing these strings eliminates some of the ambiguity of the comma and the period, since these characters are comprised in the token and are thus no longer considered as separators.

4.2 Abbreviations

Another important class of tokens incorporating the period as an element are abbreviations. Lists of abbreviations can be long and, like lists of proper names, incomplete since creation of abbreviations is a productive process. Yet their recognition is imperative for proper sentence division. Consider that in the Brown corpus there are 4819 non-unique (323 unique) abbreviations containing only letters and ending in a period. There are 48805 sentences, so taking a simplistic route and considering every period followed by a space as a final period, we would be right only for only 90% of the word-ending periods.

4.2.1 Experiment: No lexicon

We want to do better than this, of course. We can find a better approach by analyzing the structure of abbreviations. Let us consider three classes of abbreviations: A single capital followed by a period, such as *A.*, *B.*, *C.*; A sequence of letter–period–letter–period's, such as *U.S.*, *i.e.*, *m.p.h*; and a capital letter followed by a sequence of consonants followed by a period, such as *Mr.*, *St.*, *Assn.* If we automatically consider each of these sequences as abbreviations we will be right 3876 out of 3939 times in the Brown corpus. The detail is given in the table below.

regular expression	*Correct*	*Errors*	*FullStop*
[A-Za-z]\.	1323	30	14
[A-Za-z]\.([A-Za-z0-9]\.)+	626	0	63
[A-Z][bcdfghj-np-tvxz]+\.	1927	33	26
totals	3876	63	103

This means that, without consulting a lexicon, but only by using the structure of the words we will correctly recognize 3876 of the token-ending periods as part of an abbreviation (out of 4819 true abbreviations). We will introduce 63 errors by recognizing true full stops as false abbreviations, and introduce 103 ambiguities in abbreviations which should also be full stops. The number of correctly recognized sentence boundaries is then be 47696 out of 48805 (97.7%). The abbreviations in Brown that do not match the above regular expressions are the following:

etc. Fig. No. Co. Month-Names *Sen. Gen. Rev. Gov.* U.S.-State-Abbreviations *fig. Rep. Ave. Corp. figs. Figs. 24-hr. lbs. Capt. yrs. dia. Stat. Ref. Prof. Atty. 6-hr. sec. eqn. chap. Messrs. Dist. Dept. ex-Mrs. Vol. Tech. Supt. Rte. Reps. Prop. Mmes. 8-oz. viz. var. seq. prop. pro-U.N.F.P. nos. mos. min. mil. mEq. ex-Gov. eqns. dept. Yok.*

USN. Ter. Shak. Sha. Sens. SS. Ry. Rul. Presbyterian-St. P.-T.A. Msec. McN. Maj. Lond. Jas. Grev. Gre. Cir. Cal. Brig. Aubr. 42-degrees-F. 400-lb. 400-kc. 36-in. 3-hp. 3-by-6-ft. 29-Oct. 27-in. 25-ft. 24-in. 160-ml. 15,500-lb. 12-oz. 100-million-lb. 10-yr. 1.0-mg. 0.5-mv./m. 0.1-mv./m. 0.080-in. 0.025-in.

4.2.2 Experiment: No lexicon, Corpus filter

In order to reduce this list of non-recognized abbreviations without referencing a lexicon, we can use the corpus itself as a filter for identifying abbreviations. Let us define as a likely abbreviation any string of letters terminated by a period and followed by either a comma, a lower-case letter, or a number, or any string beginning with a capital letter terminated by a period.

Likely Abbreviation	*Correct (unique)*	*Errors*
[A-Za-z][∧⊔]*\.([,?]|⊔[a-z0-9]	155	81
either, not appearing without period	122	1

Using this definition of likely abbreviations recovers 138 of the 323 unique abbreviations in Brown, but introduces 54 false positives such as *chili, continuous, every, everyone, fed, feelers, finally, for, he, historians, ...* which are words that happen to end sentences that are followed by initial numbers.

We can apply the corpus itself as a filter by eliminating from the list of likely abbreviations those strings that appear without terminal periods in the corpus. This eliminates all but the word *light-hearted* from the list of false positives, and reduces the number of likely abbreviations to 122, that are validated by this filtering technique. Using the corpus as a filter for validating likely abbreviations and accepting all structures of the form [A-Z]\. or [A-Za-z]\.([A-Za-z0-9])\.)+ as non-terminal abbreviations means that 909 non-unique abbreviations remain unrecognized as abbreviations (*Mrs.* accounts for 534 of these) and are thus taken as sentence dividers, to which are added 14+63+1 = 78 cases of falsely joined sentences, thus 986 sentences are incorrectly divided giving us still a 97.9% percent recognition rate. After using the corpus as a filter, without any lexical access.

The abbreviations which are still uncaptured by this technique are the following

Mrs. No. Sept. Sen. Rev. Jan. fig. Rep. Mass. Corp. Pa. Lt. 24-hr. La. Col. Tex. Mt. Capt. in. Wash. Prof. Miss. Atty. 6-hr. eqn. chap. Ore. Mar. oz. hp. ex-Mrs. Wm. Tech. Supt. Reps. Mmes. Minn. Eq. Ed. Colo. 8-oz. seq. prop. nos. no. mos. min. mil. ex-Gov. eqns. ed. dept. al. Yok. Vs. Tenn. Sha. Sens. SS. Ry. Presbyterian-St. Pfc. OK. Mfg. McN. Maj. Kas. Jas. Ind. Eng. Del. Cmdr. Cal. Brig. App. 42-degrees-F. 400-lb. 400-kc. 36-in. 3-hp. 3-by-6-ft. 29-Oct. 27-in. 25-ft. 24-in. 160-ml. 15,500-lb. 12-oz. 100-million-lb. 10-yr. 1.0-mg. 0.5-mv./m. 0.1-mv./m. 0.080-in. 0.025-in.

As we can see in the above lists, we have a number of titles that pass through these filters. We can recuperate some titles directly from the corpus by extracting sequences of [A-Z][a-z]+\. which are followed at least twice in the corpus by a two capitalized string. When we filter these results by eliminating words that appear elsewhere in the corpus without a terminal period, we produce a list of abbreviations with no false positives, although some of the abbreviations are not properly titles: *Atty. Ave. Capt. Cmdr. Col. Dist. Dr. Drs. Gen. Gov. Jas. Lt. Mmes. Mr. Mrs. Mt. Pfc. Prof. Rep. Reps. Sen. Sens. Sr. St. Supt. Vs.* . Adding in these discovered titles reduces the number of unrecognized abbreviations to 290. and reduces the number of incorrectly recognized sentences to 368, or 1 in every 133 sentences.

4.2.3 Experiment: lexicon without abbreviations

These observations suppose that the abbreviation recognition process has no access to a lexicon. Let us examine what can be gained by using a lexicon to look up the litigious cases. Suppose now that, instead of trying to solve all the ambiguities during this tokenization phase, tokenization is reduced to number recognition and splitting words on spaces and unambiguous separators. Then every word ending a sentence as well as real abbreviations ending with the period will be sent to the morphological analyzer with a trailing period. It will then be the role of the morphological analyzer to decide if the trailing period should be isolated as a separate, sentence-ending, character. Under this supposition, the Brown corpus produces 51240 letter-initial tokens ending in a period. Suppose that we have a complete lexicon of the language in which any form of a word may be found except abbreviations and proper names. Can we discover abbreviations using this method?

We tried the following ordered filter on all strings terminated by a period:

1. if it is followed by a lower-case letter, comma or semi-colon, it becomes a known abbreviation;

2. if the word is a lower case string and the same word exists in the lexicon without a final period, it is not an abbreviation, otherwise it is an abbreviation;

3. if it begins with an upper case, and appears elsewhere in the corpus as a known abbreviation, it is an abbreviation;

4. if it begins with an upper case, and appears elsewhere in the corpus without a trailing blank, it is not an abbreviation (probably a proper name).

5. if it begins with an upper case, appears only once or twice, take it as sentence-ending word;

6. otherwise, it is an abbreviation.

The list of known abbreviations defined above contains 194 unique abbreviations such as *U.S. Jr. Mr. i.e. U.N. Co. p.m. e.g. S. a.m. Inc.* The list derived from (2) classifies most of the cases in the corpus since 42197 out of 52204 period-terminated strings fall into this class. It misclassifies *chap. fig. no. nos. prop.* and *u.* as words ending a sentence 29 times. (3) decides 78 other cases correctly, (4) decides 1827 cases but mistakes *App. Cal. Del. E. Ed. G. Jan. L. Mar. P. Rev. SS. Sept. Tech. V. W. Y.* since they appear elsewhere without a period. By the time the filter reaches the step (5), there are 563 words to consider, corresponding to 1327 instances of the 52204 period-terminated strings. This fifth step eliminates most of these instances, but incorrectly identifies the following abbreviations appearing one or two times as words: *Wm. Vol. Rte. Reps. Mmm. MMes. Eq. Aubr. Brig. Cmdr. Eng. Ind. Jas. Kas. Maj. McN. Mfg. Ore. Pfc. Pt. Rul. Ry. Sens. Sha. Vs.* and *Yok.* as words. Step (6) identifies the following as abbreviations: *Mrs. Fig. Sen. Oct. Nov. Dec. Feb. Rep. Aug. Figs. Op. Lt. Co. Pp. Mt. Capt. Wash. Ref. Prof. Atty. Stat. Schaack. Martinez. H.M.S. Christendom. Ch.* . The result of this filtering period-ending tokens through the lexicon gives the following technique. When the above filter is passed over the Brown corpus, 144 words are incorrectly tagged out of the 51240 candidates. Counting in terms of erroneously divided sentences this gives us a 99.7% success rate.

4.2.4 Experiment: lexicon with some abbreviations

Now let us suppose that our lexicon has not only all the lower-case words in the corpus, but also contains frequent abbreviations, here meaning titles (*Mr. Mrs. Dr. Sen.*), month name abbreviations (*Jan. Feb. Mar.*), U. S. state abbreviations (*Ala. Calif. Penna.*) and some common abbreviations (*etc. fig. no. Co. Ltd. Corp.*). Now the procedure will be the following, given a sequence of letters terminated by a period, it will be considered as an abbreviation: 1) if it is followed by a lower-case letter, comma or semi-colon, then it is an abbreviation; 2) if it is a known abbreviation, consider it as such; 3) otherwise, consider the word as a sentence terminator. Using the following list as a list of abbreviations in the lexicon provides us with only 53 incorrectly recognized words over the 51240 abbreviation candidates in Brown.

> Single-Letters State-Names *Assn. Av. Ave. Bldg. Blvd. Chmn. Co. Corp. Ct. Dept. Dist. Dr. Drs. Eng. Gen. Gov. Inc. Jr. Ltd. Messrs. Mr. Mrs. Msec. Mts. No. Rd. SS. Sr. St. Tech. Ter. U. USN. al. cc. cm. cu. dia. ed. etc. ft. gm. hp. hr. kc. lb. lbs. mEq. mc. mg. mil. min. ml. mm. mos. nw. pl. prop. sec. sq. var. viz. vs. yd. yrs.*

4.2.5 Related work on sentence boundary recognition

Palmer and Hearst (1994) have recently produced a technical report[3] describing an approach to sentence boundary that uses a neural net applied to morphologically tagged text to decide the case of terminal periods. They achieved a 98.5% success rate following only one minute of neural net training. Since they do not use capitalization clues, this technique might be applied to languages such as German, or to all-upper case text. In this technical report, they mention other work applied to solving this problem using regression analysis based on the individual probabilities of words appearing before punctuation(Riley, 1989), and rules based on the lexical endings of words surrounding punctuation (Müller et al., 1980).

4.3 Morphologically Analyzed Words

A major question that must be answered by the designer of the tokenizer is whether there exists a one-to-one correspondence between a token and a set of classes, or can a token correspond to a sequence of classes. For example, in the Brown corpus the word *governor's* is considered as one token and is tagged as a possessive noun. In the Susanne corpus[4] the same string is divided into two tokens *governor* and *'s* each possessing its own tag. In this case, the choice between one or two tokens seems of little importance since one would suspect that subsequent linguistic treatment would rebuild a possessive structure corresponding to that produced by one token anyway. Of greater significance is the division of *'s* in the case of strings such as *it's, he's, that's, there's, who's, she's* and with the other English contractions. If the strings are retained as one token, then the linguistic analyzer must handle the case where a single token corresponds to a sequence of tags.

The same questions must be answered for other languages. In French it must be decided whether *l'addition, m'appelle, donne-le, va-t-il, c'est-à-dire, presqu'île, tape-à-l'oeil, d'abord, ...* are to be retained as one token, or divided into many. One problem with this choice is that there are arguments to make it either way: in order to make generalizations about grammar, it would

[3]This technical report can be retrieved by anonymous *ftp* at tr-ftp.CS.Berkeley.EDU. It is in the subdirectory /pub/cs/tech-report/cds-94-797, in postscript format.

[4]Available via anonymous *ftp* at 129.67.1.165 in the directory *ota/susanne.*

be good to break out *l'* as a separate article but this introduces some ambiguity during tagging since it could also be a preverbal pronoun. A word like *rendez-vous* has possible readings as one or two tokens if the hyphen can separate words. In one case it is the noun *rendez-vous* and in the other it can be the imperative form of the reflexive verb *rendre* or the interrogative form of this verb with an inverted subject. Once the choice is made the linguistic component can take it into account, but different systems will make different choices which in turn makes comparing results or sharing tokenized text between researchers difficult. For example, available statistical tagging programs which choose parts of speech for words using their immediate context (Brill, 1992) cannot treat the case where a surface form might correspond to one or two tokens.

5 Conclusion

As we have seen, the problem of preparing raw text for a linguistic treatment raises many problems. In order to maintain as much flexibility as possible, the tokenization process should be considered as a series of modular filters through which text can be selectively passed. We have seen here that the original text file undergoes preprocessing that eliminates some markings and rejoins hyphenated words. The tokenization proper begins. One of the main purposes of tokenization is to recognize sentence and word boundaries so that lexical look-up can proceed. Certain character ambiguities can be resolved be analyzing the structure of the the input strings, in order to produce a first pass at tokenization.

Once this pass is produced, one can consider other treatments of the tokenized text before lexical lookup is performed. For example, one might consider at this point rejoining parts of a proper name separated by blanks. This can be justified as a role of the tokenizer if the space is considered as an ambiguous separator which can be disambiguated by contextual clues. In English these contextual clues are uppercase letters appearing after the first word in the sentence.

Though rarely discussed, and quickly dismissed, tokenization in an automated text processing system poses a number of thorny questions, few of which have any perfect answers.

References

Brill, E. (1992). A simple Rule-Based part of speech tagger. In *Proceedings of the Third conference on Applied Natural Language Processing*, Trento, Italy. ACL.

Francis, W. and Kucera, H. (1982). *Frequency Analysis of English.* Houghton Mifflin Company, Boston.

Müller, H., Amerl, V., and Natalis, G. (1980). Worterkennungsverfahren als Grundlage einer Universalmethode zur automatischen Segmentierung von Texten in Sätze. Ein Verfahren zur maschinellen Satzgrenzendestimmung im Englischen. *Sprache und Datenverarbeitung*, 1.

Palmer, D. D. and Hearst, M. A. (1994). Adaptive sentence boundary disambigation. Technical Report UCB/CSD 94/797, University of California, Berkeley, Computer Science Division.

Riley, M. D. (1989). Some applications of tree-based modelling to speech and language indexing. In *Proceedings of the DARPA Speech and Natural Language Workshop*, pages 339–352. Morgan Kaufmann.

Linguistic Norms and Pragmatic Exploitations or, Why Lexicographers Need Prototype Theory, and Vice Versa

PATRICK HANKS

Abstract

Dictionary compilers are constantly faced with the problem of variability in usage, which traditional linguistic theories do not account for satisfactorily. Prototype theory offers an alternative framework within which to discuss norms and variations of usage. It is suggested that dictionary entries show *meaning potentials* (rather than *meanings*), and that these meaning potentials are in fact linguistic and cognitive prototypes. Prototype theory is linked to a Gricean theory of communication and investigated in more detail. A case study in lexical analysis shows how the work of theoretical linguists on prototype theory and the work of lexicographers on empirical analysis can be complementary. A task required for successful lexical analysis in the future is the designation of semantic types and identification of the (prototypical) lexical sets that make up these types.

Problems with Received Wisdom

A modern dictionary, with its neat lists of numbered senses, offers the comforting prospect of certainty to linguistic inquirers. It suggests, "Here is a menu of choices, a list of all and only the words of the language, with all and only their true meanings. All you have to do is to choose the right one, plug it into its linguistic context, and — hey presto! — you have an interpretation". Other factors, too, encourage this view of a dictionary entry as a statement of criteria or conditions — necessary and sufficient conditions, perhaps — for the correct use of a word. 'Definitions' in most dictionaries are constructed on the principle of substitutability — that is, they are worded so as to provide a paraphrase that can be substituted for the target word in context *salva veritate* (without affecting the truth), in Leibniz's phrase (1704), or at any rate *salva interpretatione* (without affecting the interpretation). The very word 'disambiguate', so beloved of present-day semanticists and computational linguistics, implies selection from a finite menu of choices.

This is a traditional view of word meaning which goes right back (via Leibniz) to Aristotle's doctrine of essential meaning, distinguished from accidental properties.

Researchers working in artificial intelligence, machine translation, and other fields involving an empirical approach to natural-language processing and semantic interpretation have long recognized that the simplistic account of word meaning just described does not work very well in practice. Aitchison (1987, p. 49) summarizes some of the main problems thus:

> There are a small number of words such as *square* or *bachelor* which appear to have a fixed meaning, that is, they are words for which we can specify a set of necessary and sufficient conditions. The majority of words, however, do not behave in this way. They suffer from one or more of the following problems: first, it may be difficult to specify a hard core of meaning at all. Second, it may be impossible to tell where 'true meaning' ends and encyclopaedic knowledge begins. Third, the words may have 'fuzzy boundaries' in that there might be no clear point at which the meaning of one word ends and another begins. Fourth, a single word may apply to a 'family' of items which all overlap in meaning but which do not share any one common characteristic.

In my view, Aitchison concedes too much, at any rate as far as *bachelor* is concerned. The boundaries of its meaning can be shown to be quite fuzzy, especially as the word is used in present-day English society, where the institution of

marriage (on which any 'fixed-meaning' definition of *bachelor* is said to depend) is slowly being eroded by alternative lifestyles.

Be that as it may, the problem of fuzziness is more or less acute for any lexicographer trying to describe the meaning and use of almost any ordinary word of a natural language such as English. In previous papers (e.g. Hanks 1979, 1986), I have argued that dictionaries need to pay much more attention to norms of actual usage, and in 1979 I suggested that the standard theoretical view of definitions has led to some thoroughly bad lexicographical practices:

> When theory comes into lexicography, all too often common sense goes out.

By this, I meant (I now realize) traditional Aristotelian-Leibnizian theory. In those days prototype theory and frame semantics had not been invented. I went on to argue:

> Any attempt to write a completely analytical definition of any common word in a natural language is absurd. Experience is far too diverse for that. What a good dictionary offers instead is a typification: the dictionary definition summarizes what the lexicographer finds to be the most typical common features, in his [or her] experience, of the use, context, and collocations of the word.

In this paper, I shall propose that lexicography needs to temper (or replace) the received Aristotelian-Leibnizian doctrine of necessary and sufficient conditions, which may be fine for many things but is not fine for the description of natural language or human cognitive processes, with some form of prototype theory. I shall try to show how prototypes of meaning and use are associated. First, however, I need to show how word meaning and word use are rooted in Gricean communicative interaction.

Meanings and Meaning Potentials

Let us make a distinction between meanings as events and meanings as beliefs. **Meanings**, strictly speaking, are events that take place in the world, in which the participants are utterer and audience. Each participant draws on his or her mental stock of beliefs about word meaning to construct an interpretation. However, as Wright (1976) points out, "There is no guarantee, other than utterer's and hearer's common satisfaction over their mutual pragmatic success, that they are taking their meanings in the same way."

It is not surprising, therefore, that we find considerable variation in points of detail in beliefs about the true meaning of terms. Notwithstanding the difficulties, it is the shared elements in these beliefs that the unfortunate lexicographer has to try to capture in a dictionary. The job of the dictionary writer is, strictly speaking, to capture **meaning potentials** rather than meanings.

But, as Lewandowska-Tomaszczyk (1987) points out, a model of language use must reflect not only the fuzziness but also the dynamism of (shared) linguistic meaning. Language is dynamic; it should not be studied as if it were a dead thing. Founding our prototypes in the Gricean theory of conversational co-operation will help us to get the right perspective on the dynamics of meaning.

When a word is used in a text, the utterer activates some part of its meaning potential from his or her own mental store, and intends to activate a corresponding part of the hearers' or readers' mental store. The logic of this conversational correspondence has been addressed by Locke and many philosophers since, but perhaps most significantly by H. P. Grice in 1957 and 1975. Grice's papers spawned a vast literature. I do not wish to get bogged down in the niceties of Gricean theory and the nature of the 'mental store' just mentioned, but it needs to be said that the general line on conversational co-operation taken by Grice is part of the reason for wishing to distinguish meanings-as-events from the meaning potentials listed in dictionaries.

Lexicographers traditionally look at texts for evidence of the meanings of words. But of course, in looking at written texts, they see only half of the communicative story. They can see something of the utterer's intention, but they cannot see or measure the effect on the intended audience. They may say, "Well, I am part of the audience, and I know the effect on me: *I* understand the meaning of this word." This may be true, or it may be the nearest thing to the truth that a descriptive linguist can hope to achieve in practice, but it is well known that introspection is an unreliable research tool. For example, in Hanks (1990), I argued:

> Psychologically, human beings tend to register the unfamiliar rather than the familiar, the unusual rather than the usual. Thus, the files of modern dictionary publishers are full of citations for *tachograph* and *ayatollah* — words which have come into prominence within the past decade. [This was written in 1985]. But nobody notices that *take* has a common but previously unrecorded sense.

What we *think* we do when using or hearing language and what we *actually* do are not necessarily the same thing at all. A good lexicographer is always alert to the possibility that his or her own activated beliefs about the meaning of a word in context may be different in subtle ways from those of other people.

We must also bear in mind that (according to ordinary dictionaries) most words have more than one meaning. How are these to be distinguished? If a word has more than one set of meaning potentials, or if the elements of its meaning potential are grouped together in different ways to create different meanings on different occasions, then there must be some way for hearers (audience) to know which meaning potential is the right one on any particular occasion.

The simple answer is that the meaning potentials of the words that an utterer uses are projected onto the syntax. Different meanings are associated with different

syntactic patterns. Unfortunately, most dictionaries do not show clearly how this works. As we shall see, a much more delicate notion of syntax than can be found in any current dictionary will be required if we are to project meaning potentials satisfactorily onto syntax.

The advent of corpus technology, in which very large quantities of text can be stored and analysed computationally, is now enabling researchers to observe that ordinary language in use is very much more highly patterned than predicted either in traditional dictionaries or in most linguistic theory. Mainstream linguists and lexicographers up to now have asked questions about what is possible: "Can you say this in English?" All too often, the answer is, "Well, yes, I suppose you could... but it wouldn't be normal."

Corpus linguists today tend to ask a different kind of question. They are concerned with performance rather than competence, and they are concerned with norms rather than possibilities. They ask, for example, "Is it normal to say this in English?"

The question is of fundamental importance, not only to language teaching, but also to an understanding of such phenomena as literary style, poetic usage, mannered writing, metaphor, and meaning change, not to mention actual errors. Before you can know how a linguistic convention can be exploited, you need to be able to say what the convention is. Before you can account for unusual meanings of a word, you have got to be able to say what the ordinary meanings are. It is probably not too much of an exaggeration to say that, at the lexical level, the conventions of English (or any other language) have not yet been satisfactorily described by anyone. At the same time, contemporary dictionaries contain many unnecessary postulated senses, simply because the lexicographers have failed to achieve an appropriate level of generalization or have presented syntactic distinctions as if they were semantic ones. Corpus evidence also shows that some important, everyday, conventional uses have been completely overlooked by dictionaries.

Projecting Meaning Potentials onto Syntax

Let me give an example designed to show how the meaning potential of a word projects onto the syntax and why the level of syntactic description needs to be extremely delicate. It is generally agreed that verbs, being the pivots of clauses, have certain grammatical structures — subject, object, and adverbial — associated with them, which are linked to their meaning. So, for example, the meaning of the verb *bank* differs depending on its transitivity. But we also need to say something in the syntax about the semantic type of its subject and object. An aircraft banks (intransitive); people bank money (transitive); a pilot banks an aircraft (also transitive, but the semantic type of *aircraft* is very different from the semantic type of *money*). These two facts (the verb's transitivity patterns and the semantic types of its arguments) determine the way in which we interpret it.

The patterns just mentioned may be exemplified in the following four sentences, taken from the British National Corpus. I have shown a partial parsing for the sentences, including an indication of relevant semantic types.

1. [SUBJ Jani [HUMAN]] banked-VBD [OBJ £60,000 [MONEY]] through successful libel actions against Options magazine and the London Evening Standard.

2. . . . [SUBJ she [HUMAN]] is believed to have banked-VBN [OBJ £10_million [MONEY]] since being booted out of Downing Street two years ago.

3. [SUBJ The plane [VEHICLE]] banked-VBD [NO OBJ], and he pressed his face against the cold window.

4. [SUBJ I [HUMAN]] banked-VBD [OBJ the aircraft [AIRCRAFT]] steeply and turned.

To account for sentences such as these, the relevant parts of the accompanying entry for *bank* in a formal dictionary entry, showing how the meaning potential projects onto the syntax, would be something like this:

5. [SUBJ[HUMAN]] ___ [OBJ[MONEY]]
= deposit or invest [MONEY] in a bank or other financial institution for safe keeping

6. [SUBJ[AIRCRAFT]] ___ [NO OBJ]
= raise one wing higher than the other in order to change direction

7. [SUBJ[HUMAN]] ___ [OBJ[AIRCRAFT]]
= cause [AIRCRAFT] to raise one wing higher than the other in order to change direction

Semantic types remain to be identified and listed, in the form of lexical sets. If an accurate a-priori description of semantic types were possible, then the semantic types of language would be as familiar to us as the well-established part-of-speech classes: verb, noun, adjective, etc. In a sense, they are: joking aside, we all know, informally, that the expression "Mrs Thatcher" falls into the class [HUMAN]. But a satisfactory formal account of such classes is not yet available. Preliminary empirical work suggests that all a-priori assumptions are suspect. For example, the class [HUMAN] seems plausible enough, but it may turn out to be unsatisfactory. As a matter of syntax, it may work better if divided into two classes: defined, on the one hand, by properties which Mrs Thatcher shares with cats, horses, and monkeys, such as eating, sleeping, and climbing [i.e. the type ANIMAL], and on the other hand by properties which she shares with nations, governments, business organizations, family-history societies, and computers [i.e. the type COGNITIVE], namely analysing, negotiating, banking money, making statements, expressing sympathy, and so forth. The details at present are uncertain, so for present purposes I shall continue to use [HUMAN].

We cannot rule out the possibility that the relevant set of semantic types for each verb in a language will turn out to be slightly different from those relevant to every other verb. But it is to be hoped that at least some gross overlaps will be discovered, e.g. that there are many features common to the direct objects of, say, causative verbs of motion or verbs of perception. These overlaps remain to be established as an empirical fact. One thing is already clear, however: the sets of semantic types are extremely fuzzy. If they can be identified at all, set membership will be stated in terms of similarity to a contextually determined prototype. Interestingly enough, it seems likely that statistical tests of the kind described in Church, Gale, Hanks, *et al.* (1990, 1994) may help us to identify set membership.

Prototype Theory

If the numbered senses in dictionaries can be seen as lists of meaning potentials rather than meanings, then it is only a short step to arguing that meaning potentials are in fact prototypes. If we take this step, we can draw on the rich literature on prototype theory that has grown up since Eleanor Rosch delineated the notion of conceptual prototypes in 1978. Probably the best account of prototype theory for present purposes is to be found in Taylor (1989). At the core of his book (pp. 59, 60), Taylor says:

> The prototype can be understood as a schematic representation of the conceptual core of a category. . . .
>
> Entities are assigned membership in a category in virtue of their similarity to the prototype; the closer an entity to the prototype, the more central its status within the category.

Moreover (p. 61),

> Prototypicality is recursive, in that the very attributes on whose basis membership in a category is determined are more often than not themselves prototype categories.

This, then, is a foundation on which to build. The ground floor of our theory for lexicography will consist of syntactic prototypes; the upper floors will consist of cognitive categories and prototypes as explored by writers such as Rosch (1978) and Lakoff (1987), and the roof (if this is not extending the metaphor too far) will be stereotypes as in Putnam (1975), including "the division of linguistic labor", favouring technical expertise over folk knowledge for certain kinds of meaning potentials.

If lexicographers are to take prototype theory seriously as a theoretical foundation for their work, they will need to be very clear about the distinction between sociolinguistic and psycholinguistic prototypes. All speakers of a language rely unconsciously on their belief in the existence of shared conventions of meaning as well as syntax in order to achieve successful communication. To the extent

that a language is learned by its native speakers (rather than pre-programmed genetically), each speaker acquires it differently and in different ways. In the words of Quine (1960),

> Different persons growing up in the same language are like different bushes trimmed and trained to take the shape of identical elephants. The anatomical details of twigs and branches will fulfil the elephantine form differently from bush to bush, but the overall outward results are alike.

This is the psycholinguistic prototype: largely a prototype of belief. The sociolinguistic prototype is rather different: it is a syntactic pattern, a pattern of linguistic behaviour which can be identified by painstaking corpus analysis. A corpus provides only indirect evidence for word meaning. A large element of interpretation is required to get from the **syntactic patterns** observed in a corpus (traces of linguistic behaviour) to **meanings**, and a further interpretative step is required to get from **meanings** to **meaning potentials**. The task of the ideal lexicographer is to show how each meaning (in an ideal dictionary) is associated with one or more syntactic patterns, which we may regard as prototypes of linguistic behaviour.

Consequences of Prototype Theory for Lexicography

What are the consequences if we invoke prototype theory to explain dictionary definitions? It is clear there are advantages, but what are the disadvantages?

An objection sometimes voiced is that prototype theory involves abandoning the certainties of Aristotelian conceptual categories and replacing them with something that is so uncertain as to be almost meaningless. It is undoubtedly true that adoption of prototype theory leads to abandoning comfortable certainties. However, it seems that prototypes stand a better chance of being true, since meaning potentials are themselves vague and variable. In the words of Anna Wierzbicka (1985):

> An adequate definition of a vague concept must aim not at precision but at vagueness: it must aim at precisely that level of vagueness which characterizes the concept itself.

A good prototype will not be so vague as to be meaningless, but will show quite precisely the combinations of conventional syntactic and semantic features that go to make up the conventional usage and meaning potential of a word. If we then find uses in a text in which none of the conventional features are present or can be activated, they are mistakes. When only some of the features are present, we may be looking at a mistake, or we may have found some kind of literary or metaphorical exploitation. Prototype theory provides a machinery for talking about the great area between correct and incorrect.

Most monolingual lexicographers since 1755 (when Dr Johnson's great dictionary of the English language was published) have been straining after the idea that

they were constructing sets of necessary and sufficient conditions for "correct" word meaning. So the idea that what they are actually doing is constructing prototypes would necessitate a radical revision of many entries. Most of the rest of this paper is devoted to exploring what might be entailed in such a revision.

We are contrasting a theoretical tradition of approximately 20 years' duration with one of approximately 2400 years' duration. So far, only one major dictionary (The Collins Cobuild English Language Dictionary, ed. Sinclair, Hanks, *et al.* 1987) has come anywhere near taking systematic notice of prototype theory. Sinclair's work is, in part, founded on that of Halliday, as for example (1966), where he argues against making too sharp a distinction between grammar and use of language, and proposes to "supplement the grammar by formal statements of lexical relations". There is a still a lot of work to be done in this area.

To illustrate the sort of work that is needed, building on the start made by Cobuild, I shall offer a case study showing how the meaning potential of a word is associated with its syntax, in the light of prototype theory. The word chosen is *climb*, partly because it has been much discussed in the literature, and partly because it was one of the words studied in detail as part of the 'Hector' project, a collaboration between Oxford University Press and the Systems Research Center of Digital Equipment Corporation, described by my colleague Sue Atkins at the 1992 Complex conference (Atkins 1992).

Climb: Theoretical Analysis

One of the questions we asked ourselves in the course of the Hector project was: Is there a better model than a list of numbered definitions for representing the tenuous interplay of norms and variations by which words in use make meanings?

Any case study in lexical analysis should start by reviewing the theoretical basis, then go on to analyse the data, seeking theoretically sound ways of accounting for the linguistic patterns and constructions that may be found. This is how I shall proceed.

To review the theoretical literature, we may start with Fillmore (1982), who goes to the heart of things:

> Semantic prototypes can be realized in at least six ways, named here by the typical English words which exemplify them. . . .
>
> *Case 1: Type CLIMB* The category is identified in terms of a disjunction of mutually compatible conditions, and the best examples are those in which all members of the disjunction are present.
>
> The English verb *climb* can be taken in illustration of Case 1. Its two critical conditions may be named Clambering and Ascending. A monkey climbing up a flagpole satisfies both of these and thus exemplifies the

> prototype well. A monkey clambering down a flagpole, or clambering horizontally in the rafters of a warehouse, can also be said to be climbing, even though in that case only the Clambering component is present. A snail ascending a wall, in the way a snail usually moves, can be said to be climbing (up) the wall, even though in that case only the Ascending component is present. (Snails, lacking limbs, cannot clamber.) But the snail when returning to the bottom of the wall cannot be described as climbing, since it is neither ascending nor clambering. Either of the two critical conditions may be absent; but they may not both be absent.

To Fillmore's account, we need to add the notion of preferences. Preference semantics was invented by Wilks (1975), who says in a seminal paper:

> The key point is that word sense and structural ambiguity in natural language will always, in any system, give rise to alternative competing structures, all of which can be said to 'represent' whatever chunk of natural language is under examination. What I mean by 'preference' is the use of procedures, at every level of the system, for preferring certain derived structures to others, on the basis of their 'semantic density'.

Wilks shows how the notion of semantic preference may be used to resolve problems of anaphoric reference, for example identifying the antecedent of *it* in fragments such as:

> 8. John left the window and drank the wine on the table. It was good.

vs.

> 9. John left the window and drank the wine on the table. It was brown and round.

The formulation of a preference rule system in Jackendoff (1990) makes clear the relevance of this notion for our present purpose. In part I of that book, headed 'Basic Machinery', he says:

> **Preference Rule System**
>
> Consider the following examples:
>
> 10a. Bill climbed (up) the mountain.
> 10b. Bill climbed down the mountain.
> 10c. The snake climbed (up) the tree.
> 10d. ?* The snake climbed down the tree.
>
> *Climbing* appears to involve two independent conceptual conditions: (1) an individual is travelling upward; and (2) the individual is moving with characteristic effortful grasping motions, for which a convenient term is *clambering*. On the most likely interpretation of (a), both these conditions

> are met. However, (b) violates the first condition, and, since snakes can't clamber, (c) violates the second. If *both* conditions are violated, as in (d), the action cannot at all be characterized as climbing. Thus neither of the two conditions is necessary, but either is sufficient.
>
> However, the meaning of *climb* is not just the disjunction of these two conditions. That would be in effect equivalent to saying that there are two unrelated senses of the word. If this were the correct analysis, we would have the intuition that (a) is as ambiguous as *Bill went down to the bank.* But in fact it is not. Rather, (a), which satisfies both conditions at once, is more 'stereotypical' climbing. Actions that satisfy only one of the conditions, such as (b, c), are somewhat more marginal but still perfectly legitimate instances of climbing. In other words, the two conditions combine in the meaning of a single lexical item *climb*, but not according to a standard Boolean conjunction or disjunction. [Jackendoff (1983)] calls a set of conditions combined in this way *a preference rule system*, and the conditions in the set *preference rules* or *preference conditions*.

Both Wilks and Jackendoff posit a further aspect of preference rule systems, namely that when one lacks information about the satisfaction of the conditions, they are assumed to be satisfied as *default values*. Thus, "The reason why (10 a) is interpreted as stereotypical climbing is that the sentence gives no information to the contrary. It is only in the (b) and (c) sentences, which do give information to the contrary, that a condition is relinquished." (Jackendoff 1990, p. 36)

Jackendoff actually goes further, seeking to supplement "feature-based semantics" in conceptual analysis with a "three-dimensional model" of a word's meaning. (Actually, a four-dimensional model, since verbs of motion also involve the dimension of time.) Lexicography cannot follow him here, however, nor is there any need to until or unless multimedia language reference tools replace traditional dictionaries. For lexicographic purposes, the use of words to express conceptual structures is inevitable. This will inevitably look like a feature analysis with semantic components, but, as we shall see, the status of those components is preferential and probabilistic, rather than necessary.

Wierzbicka (1990) comments on Jackendoff's analysis:

> But this analysis is deficient . . . because it fails to predict, for example, that if a train went quickly up a hill it couldn't be described as 'climbing'. There is a difference in meaning between the (a) and (b) variants in the following pairs of sentences:
>
> (a) The train climbed the mountain.
> (b) The train shot up the mountain.
>
> (a) The temperature climbed to 102 degrees.
> (b) The temperature shot to 102 degrees.

> Despite his rich arsenal of descriptive devices, including multiple brackets and 'preferential features', Jackendoff's analysis cannot account for facts of this kind.
>
> In my view, all that is really needed to account for such facts is a more careful, and more imaginative, phrasing of the necessary and sufficient components of the concept 'climb'. Tentatively, I would propose the following:
>
> > X climbed. . . = X moved like people move in places where they have to use their arms and legs to move upwards
>
> interpreted as referring to anything other than slowness. For trains, it can be interpreted as referring to slowness and apparent difficulty. For people, too, it can be interpreted as referring to slowness and apparent difficulty; but it can also be interpreted as referring to a quick and apparently effortless movement upwards in places where normally people would have to use their arms and legs to move upwards at all (cf. 'Watching him climb the cliff quickly and effortlessly, I was filled with pride and admiration').
>
> Thus, a prototype is indeed relevant to the concept 'climb'. But this prototype is not 'suppressed' in less typical uses of the verb. It is part of the semantic invariant itself.

It will be clear from what I have said so far that I believe Wierzbicka's attempt to rescue necessary and sufficient conditions to be doomed. For one thing, it relies on the word 'like' in her definition of *climb*. This reduces the necessary condition to one that is trivially true. It will be true whatever is said, for as Davidson (1978) points out, "All similes are true and all metaphors are false. . . . everything is like everything else."

Lexical analysis in prototype theory, then, will draw heavily on the notions of preferences and default values. A third, equally important concept mentioned by Jackendoff is:

> a repertoire of major conceptual categories, "the semantic parts of speech". These categories include such entities as Thing (or Object), Event, State, Action, Place, Path, Property, and Amount.

Jackendoff's "semantic parts of speech" sound quite similar to Hanks's "semantic types", described above. A point of difference is that I propose that semantic types should be discovered empirically and arranged into lexical sets whose typical members are to be listed rather than assumed a priori.

The question arises, how confident can we be in relying on "semantic parts of speech" in our analysis? After all, the major syntactic parts of speech (noun, verb, etc.) have been pretty well established for around 2000 years in European grammatical theory. How come the so-called "semantic parts of speech" are not

equally well established? Perhaps it is because, unlike regular part-of-speech classes, they are extremely fuzzy sets, of a kind which could not stand up at all until fuzzy logic was invented (Zadeh 1965). Actually, as Geoffrey Sampson (1987) has pointed out, the traditional part-of-speech classes are also open-ended and fuzzy, but they are so large and have so many central and typical members that the fuzziness was long thought to be an irritating side-issue rather than a central property of the class.

Climb: Empirical Analysis

It is not the primary purpose of theoretical discussions such as the foregoing to improve individual entries in ordinary dictionaries. However, if we look at ordinary dictionary entries (examples are given in Figure 1), we can see that this would be a beneficial side effect. Neither of these works have identified the 'CLAMBER' meaning sense of *climb* as succinctly as Fillmore and Jackendoff, though both dictionaries are obviously troubled by contexts in which the 'ASCEND' component is absent.

But, as the quotation from Wierbicka shows, theoretical analysis alone can leave haunting doubts. How serious is the threat to necessary and sufficient conditions from counterexamples? Are such counterexamples central or peripheral? Are problems with the traditional account of word meaning hopelessly flawed, or are the differences merely a matter of taste? What is the status of uses not accounted for by the theoretical account? Has some important or central component of the prototype been overlooked entirely, or are we dealing merely with boundary cases?

Thoroughgoing empirical analysis of a well-selected corpus can go a long way to resolve those doubts. The aim of an analysis such as that shown in Figure 2 is to account for all and only the conventional uses of the verb, while at the same time showing how it varies according to context. There is only one prototype for *climb*, since all the features are related in a Wittgensteinian family resemblance. Other verbs (for example, *bank*, where there is a disjunction of features) may have more than one prototype. If the analysis has been done properly, any use of the English verb *climb* not accounted for in Figure 2 is either an exploitation (literary trope, metaphor, etc.) or a mistake. A separate set of rules needs to be compiled to show how the prototype may be exploited.

Figure 2 records both the core facts about the meaning potentials of *climb* and the way in which these vary according to context. Like all verbs, the valency slots around *climb* attract some lexical items more strongly than others, and these can be summarized in the form of lexical sets, which are sets of default preference conditions. The analysis in Figure 2 is supported by a selection of corpus evidence, given in Figure 3. Figure 3 also includes a few examples of exploitations of conventions, illustrating metaphors and boundary cases for correct usage.

The salient features of the analysis are as follows. The headings show subject,

direct object, and adverbial complement slots surrounding the verb itself. Most of the items in capital letters are themselves prototypes (remember, prototypes are recursive, and explained in terms of other prototypes). The meaning potential of the verb in each of these prototypical contexts is shown in the column headed 'V'. Meaning potentials that derive from the combination of elements rather than from any single component are in the column headed 'comb.'

At the highest level of analysis, we note that *climb* appears in four syntactic patterns:

1. with a direct object
2. with a null complement
3. with no direct object and an adverbial complement
4. with an abstract subject, an amount as optional direct object, and an optional adverbial complement also involving an amount

Let me now give a few more detailed informal comments on each of these patterns.

Pattern 1: At the heart of the *climb* prototype are uses in which the subject is human and the direct object is a thing such as a mountain, building, tree, barrier, stair, or path. The subject may also be an animal or even a vehicle. If the direct object is a mountain or building, there is a prototypical implication that climbing it results in the climber getting to the top. This is not so if the direct object is a tree. If the direct object is a barrier, such as a style or wall, there is a strong or weak implication that the climber goes up and over. Generally, the combination of subject, verb, and object imply that the climber uses all his or her limbs, but if the direct object is a staircase or path, the climber proceeds on foot. Obviously, if the subject is a vehicle, it has no limbs to use, so it proceeds in its normal way, namely on wheels, and the direct object will be a path (not a mountain, building, tree, barrier, or stair). Finally, if the subject is a path or road and the verb is transitive, the object will be another word in the same set, as in 11.

11. The smaller unpaved road climbed a shallow hill . . .

If the subject is a path or road, the categorization of the verb changes from event to state.

Adverbial phrases are sometimes found complementing this pattern, but if so they are to be regarded as optional extras, not part of the prototype as in pattern 3.

Pattern 2: *climb* also occurs in a null-object alternation. Here, in the most central use, the default interpretation 'suppressed direct object: mountain' is subsumed, as in 12.

12. Harlin began to climb.

Sinclair (1990, p. 49) calls such uses 'text-transitive'. Discussing the verb

decline, he points out that "Whatever is declined is expressed in the text in one way or another" (i.e. other than as an overt direct object).

With this pattern, we also place uses such as those in 13 and 14.

13. Oily smoke was climbing from the burning trucks.
14. The sun was climbing into a cloudless sky.

For such sense, there is an optional adverbial complement. Here we are near the boundaries of the prototype, and the pattern is very unstable. For example, the semantic component 'THROUGH AIR' may be moved out of 'comb.' and made explicit, as in 15.

15. We climbed through the cloud which had now formed.

Many people would say that such uses are metaphorical and should therefore be classed as exploitations rather than as part of the prototype. However, since they are conventionalized metaphors if they are metaphors at all, it seems better to include them here.

Pattern 3: The subject is human or animal and there is an adverbial complement, usually involving a prepositional phrase (*from* here, *through* there, *under* that, *to* there). This is the celebrated sense which seems to have caused some perplexity for our traditional dictionaries, in which *climb* has no sense of 'go upward', but rather only a sense of 'go with effort'. However, if the subject is a path, as in 16, then the conditions associated with PATH at pattern 1 apply: the verb is a verb of state and the meaning is 'UPWARD'.

16. A precipitous road climbs from Batcombe to the crest of the downs.

Pattern 4: The subject is something abstract such as prices or temperature, and the meaning is 'become greater' or rise on a scale. There are optional adverbial complements expressing the amount by which something becomes greater and/or the level that it reaches, as in 17.

17. The MIB climbed 10 points to 1088.

Many more comments could be made, but those are the main points. The representation may seem complex, but actually it is quite straightforward. It would, however, benefit greatly from a hierarchical three-dimensional presentation, as would be possible in a hypertext on-line dictionary such as that described in Atkins (forthcoming), rather than the flat, two-dimension presentation given here.

Conclusion

The example of *climb*, discussed exhaustively here, suggests that there is a need

for detailed empirical analysis of the lexicon, projecting the meaning potentials of words onto the syntactic patterns with which they are associated. *Climb* is actually one of the simpler verbs of motion: its analysis may serve as a model when more complex items are tackled. There are between 5000 and 8000 verbs in English which demand analysis in this way. Nouns and adjectives demand rather different treatment — but that is a subject for a different paper.

We have seen how theory-based analysis can interact with empirical analysis, to the benefit of both. The empirical analysis itself demonstrated how some elements in the meaning potential of a word are associated with combinations rather than with individual lexical items.

Above all, we saw how various lexical sets in particular syntactic roles can alter the meaning of the target word. For this reason, it is particularly unfortunate that work on semantic types and lexical sets is in a rather primitive state. There is an urgent need, it seems to me, for a list of the lexical sets that are relevant clues for selecting the appropriate meaning of each other word in the language. These lexical sets may be designated as semantic types, but they play a syntactic role. They are fuzzy sets, and they are themselves prototypical in character.

* * * * * * * * * * *

Bibliography

Aitchison, Jean (1987): *Words in the Mind* (Blackwell)

Atkins, B. T. S. (1992): 'Tools for Computer-Aided Corpus Lexicography: the Hector Project' in *Proceedings of Complex 92* (Hungarian Academy of Sciences)

Atkins, B. T. S. (forthcoming): 'Word Meaning as a Gestalt: A Frame-Semantics Approach to Lexical Description'

Church, K., W. Gale, P. Hanks, and D. Hindle (1990): 'Using Statistics in Lexical Analysis' in U. Zernik (ed.), *Lexical Acquisition: Using on-line Resources to Build a Lexicon* (Lawrence Erlbaum Associates)

Church, K., W. Gale, P. Hanks, D. Hindle, and R. Moon (1994): 'Lexical Substitutability' in B. T. S. Atkins and A. Zampolli (eds.), *Computational Approaches to the Lexicon* (Oxford University Press)

Davidson, Donald (1978): 'What Metaphors Mean' in *Critical Inquiry*, vol. 5

Fillmore, Charles J. (1982): 'Towards a Descriptive Framework for Spatial Deixis' in R. J. Jarvella and W. Klein (eds.), *Speech, Place, and Action* (John Wiley and Sons)

Grice, H. P. (1957): 'Meaning' in *Philosophical Review*, vol. 66.

Grice, H. P. (1975): 'Logic and Conversation' in P. Cole and J. L. Morgan (eds.), *Syntax and Semantics, vol. 3: Speech Acts* (Academic Press)

Halliday, M. A. K. (1966): 'Lexis as a Linguistic Level' in C. E. Bazell, J. C. Catford, M. A. K. Halliday, and R. H. Robins (eds.), *In Memory of J. R. Firth* (Longman)

Hanks, Patrick (1979): 'To What Extent does a Dictionary Definition Define?' in R. R. K. Hartmann (ed.), *Papers from 1978 B.A.A.L. Seminar on Lexicography* (Exeter Linguistic Studies)

Hanks, Patrick (1986): 'Typicality and Meaning Potentials' in M. Snell-Hornby (ed.): *ZuriLEX '86 Proceedings*

Hanks, Patrick (1990): 'Evidence and Intuition in Lexicography' in Jerzy Tomaszczyk and Barbara Lewandowska-Tomaszczyk (eds.), *Meaning and Lexicography* (John Benjamins Publishing Company)

Jackendoff, Ray (1983): *Semantics and Cognition* (MIT Press)

Jackendoff, Ray (1990): *Semantic Structures* (MIT Press)

Lakoff, George (1987): *Women, Fire, and Dangerous Things* (The University of Chicago Press)

Leibniz, G.W. von (1704): 'Table de définitions' in L. Couturat (ed., 1903) *Opuscules et fragments inédits de Leibniz* (Paris)

Lewandowska-Tomaszczyk, Barbara (1987): *Conceptual Analysis, Linguistic Meaning, and Verbal Interaction* (Acta Universitatis Lodziensis)

Putnam, Hilary (1975): *Mind, Language, and Reality: Philosophical Papers. Volume 2* (Cambridge University Press)

Quine, Willard Van Orman (1960): *Word and Object* (MIT Press)

Rosch, Eleanor, and B. B. Lloyd (1978): *Cognition and Categorization* (Lawrence Erlbaum Associates)

Sampson, Geoffrey (1987): 'Evidence against the 'Grammatical/Ungrammatical' Distinction' in Willem Meijs (ed.) *Corpus Linguistics and Beyond* (Rodopi)

Sinclair, John, Patrick Hanks, *et al.* (eds., 1987): *Collins Cobuild English Language Dictionary* (Harper Collins)

Sinclair, John (1991): *Corpus, Concordance, Collocation* (Oxford University Press)

Taylor, John R. (1989): *Linguistic Categorization: Prototypes in Linguistic Theory* (Oxford University Press)

Wierzbicka, Anna (1985): *Lexicography and Conceptual Analysis* (Karoma, Ann Arbor)

Wierzbicka, Anna (1990): 'Prototypes save: on the Uses and Abuses of the Notion of "Prototype" in Linguistics and Related Fields' in Savas L. Tsohatzidis (ed.) *Meanings and Prototypes: Studies in Linguistic Categorization* (Routledge)

Wilks, Yorick (1975): 'A Preferential, Pattern-Seeking Semantics for Natural Language Inference' in *Artificial Intelligence*, vol. 6

Wright, Edmond L. (1976): 'Arbitrariness and Motivation: a New Theory' in *Foundations of Language*, vol. 14.

Zadeh, Lotfi (1965): 'Fuzzy Sets' in *Information and Control*, vol. 8

climb /klʌɪm/ *v.* Pa. t. & pple **climbed**, (*arch.*) **clomb** /kləʊm/. Pa. t. also †**clamb**. [OE *climban* = (M)LG, (M)Du. *klimmen*, OHG *klimban* (G *klimmen*), f. WGmc nasalized var. of base of CLEAVE *v.*[2] (orig. = hold fast).] I *v.i.* 1 Raise oneself by grasping or clinging, or by the aid of hands and feet; ascend a steep place. Freq. foll. by *up* (adv. & prep.). OE. b Rise with gradual or continuous motion; (of the sun, an aeroplane, etc.) go upwards, move towards the zenith; *fig.* increase steadily. OE. c *fig.* Rise in dignity, rank, or state by continued effort; ascend in the intellectual, moral, or social scale. ME. d Of a plant: creep up by the aid of tendrils or by twining. LI8. 2 Slope upwards. ME. 3 Foll. by *down*: (*a*) (adv. & prep.) lower oneself (along) by grasping or clinging, or by the aid of hands and feet; (*b*) *fig.* (adv.) withdraw, esp. with ignominy, from a position taken up, abandon a declared position. ME.

NEW SHORTER OXFORD ENGLISH DICTIONARY

climb /klaɪm/ *v.* & *n.* —*v.* **1** *tr.* & *intr.* (often foll. by *up*) ascend, mount, go or come up, esp. by using one's hands. **2** *intr.* (of a plant) grow up a wall, tree, trellis, etc. by clinging with tendrils or by twining. **3** *intr.* make progress from one's own efforts, esp. in social rank, intellectual or moral strength, etc. **4** *intr.* (of an aircraft, the sun, etc.) go upwards. **5** *intr.* slope upwards.

CONCISE OXFORD ENGLISH DICTIONARY

fig. 1: Some dictionary definitions for the verb climb

S	V	comb.	O	A
Thing $_i$ LS: HUMAN LS: ANIMAL	Event GO UPWARD WITH EFFORT SLOWLY (?)	USING ALL LIMBS TO TOP OF (?)	Thing $_j$ LS: MOUNTAIN LS: BUILDING	
		USING ALL LIMBS	LS: TREE LI: ladder LI: drainpipe LI: scaffolding	
		USING ALL LIMBS UP AND OVER	LS: BARRIER	
		ON FOOT	LS: STAIR LS: PATH	
LS: VEHIC	GO UPWARD SLOWLY	ON WHEELS	LS: PATH	
LS: PATH$_i$	UPWARD	State	LS: PATH$_j$	
LS: HUMAN	Event GO UPWARD WITH EFFORT SLOWLY	UP MOUNTAIN USING ALL LIMBS	0	
LS: PLANE	GO UPWARD	THROUGH AIR	0	(ADVERBIAL from SOURCE to GOAL)
LS: VAPOUR	GO UPWARD	THROUGH AIR	0	(ADVERBIAL from SOURCE to GOAL)
LI: sun	GO UPWARD PERCEIVED		0	
LS: PLANT	GROW UPWARD	AROUND THING	0	
LS: HUMAN LS: ANIMAL	Event GO WITH EFFORT	USING ALL LIMBS	0	ADVERBIAL from SOURCE via PATH to GOAL
LS: PATH	UPWARD	State	0	ADVERBIAL from SOURCE to GOAL
LS: ABSTRACT	Event BECOME GREATER		(AMOUNT)	(ADVERBIAL by AMOUNT to AMOUNT)

LS = lexical set; LI = lexical item

fig 2: Prototype for `Climb'

CLIMB EXAMPLES

I. TRANSITIVE USES

[HUMAN] climb [THING]

1. Stalin died in 1953, and Hillary climbed Everest 'because it was there". In
2. dge University Climbing Club, to climb Mont Blanc by the Goutier route befo
3. r walkers, and almost anyone can climb Triglav: the last refuge is only 400
4. road range. When Charles Whitman climbed the university tower in Austin, Te
5. Wood Green School, Witney. They climbed a drainpipe to enter the school th
6. lete that the postman has had to climb a ladder to the front entrance to de
7. d generously collusive. He could climb an oak and sit there alone for all o
8. nted it. Show her a tree and she climbed it. Not so Prince Charles. He was
9. earsing everything. If necessary climb the scaffolding yourself to get the
10. ur climb. Young boys are forever climbing things." Beaming she swung the ga

****

11. ion. How good are the beetles at climbing cereal plants and locating aphid
12. don't know whether to eat it or climb it!" A five-minute drive up the roa
13. med down into the troughs before climbing the next steep wave. Away from th
14. plotter in the Air Force before climbing the civil service ladder with a j
15. he answer is probably that he is climbing the ladder of a lucrative career

****

16. the end of the footpath and then climbed a stile. He believed he dot home u
17. 1942 I should think, I remember climbing some railings at the back of Guil
18. g refugees. Some of the refugees climbed the embassy wall. Others broke thr
19. conceived of the possibility of climbing the Abbey wall. Now suddenly it s

****

[HUMAN] climb [STAIR]

20. we crawled, troglodytes all. We climbed a narrow and broken staircase towa
21. ago on a gentle Autumn evening I climbed some steep stairs in a converted h
22. xiety. She chewed her lip as she climbed the remaining stairs to Nevil's do
23. g; a rectilinear spiral. She had climbed nearly 400 steps and

****

[HUMAN] climb [PATH]

When the direct object is a PATH word, it is not always clear from the immediate context whether the subject refers to people on foot or in a vehicle. In this case, the condition "ON_FOOT" may nevertheless be taken as applicable: it is a default, so it applies until and unless it is overridden by evidence to the contrary.

24. d through the ford and began to climb the gradual slope beyond. dogs barke
25. ce. It was still raining as we climbed the pass to the Spanish frontier,
26. er water seemed louder when she climbed the road by herself. Martha though
27. of hundred feet above as they climbed the slope, like a fortress behind

fig. 3

28. gaps in the teak boards as we climbed the gangplank. A plump old man sit
29. Rashidiyeh. But they had never climbed the hill. There are, of course, s

****

[PATH] climb [PATH]

Expresses a state rather than an event

30. hamlet the smaller unpaved road climbed a shallow hill before disappearing
31. tray of refreshments. The lawn climbs a slope several yards in front of t
32. down to Boscombe Pier. It then climbs the inevitably steep hill back up t

****

[VEHICLE] climb [PATH]

33. were bumper to bumper as they climbed Headington Hill, the Astra behind
34. very efficiently. A trolleybus climbing a hill was often aided by power f

---

II. NULL COMPLEMENT

[HUMAN] climb

35. gainst the rock, Harlin began to climb. Charsky stared up after him. Then s
36. a mixed Italian and German team climbing not far away, heading for

****

[PLANE] climb

37. where it was grown." The plane climbed ponderously but the mountain slid
38. outh overhead Dunster Castle we climbed through the cloud which had now fo

****

[VAPOUR] climb

39. th the column of steam and ash climbing eleven kilometres high above the
40. h explosions and oily smoke was climbing from the burning trucck ks to t
41. her than later. Thunder-clouds climbed steeply over Poitiers, and as Peli

****

sun climb

42. matched their joy; the sun was climbing into a cloudless sky and beginnin
43. . But faces grew red as the sun climbed, the cicadas chanted and the tar b

---

fig. 3 cont.

III. WITH PP COMPLEMENT

[HUMAN] climb [from SOURCE] [via PATH] [to GOAL]

44. to a halt in front of her Maggie climbed aboard and went upstairs. She ador
45. aded when approaching a house or climbing across a fence. If it hadn't been
46. the embassy railings even as she climbed across to safety. Only the interve
47. olice said the man was trying to climb from a tower block's seventh floor t
48. ket. Angry workers glowered as I climbed from my car. A policeman waved me
49. Taylor said: 'We have a man who climbs in with the sharks to clean the tan
50. Charlie loaded up the van, then climbed in. 'Mr Lawler will be upset that
51. to Mum and Dad's room. There he climbs into bed and goes to sleep. Mum and
52. fice in Sanaya, west Beirut, and climbed into his armoured Mercedes, waving
53. limbing-frame. That it should be climbed on, into and through, compliments
54. The front door blocked, the men climbed onto the roof and then things got
55. slowly, Gower wandered back and climbed over the stile. He made wretchedly

****

[PATH] climb [ADVERBIAL OF DIRECTION]

Expresses a state rather than an event

56. and verges. A precipitous road climbs from Batcombe to the crest of the d
57. ery here is in perfect order. It climbs in tiered rows up a hard, bare hill
58. oot, banks thick with daffodils, climbing out of sight, 'She would enjoy t
59. e next mile is a wonderful walk, climbing out of the valley, with panoramic
60. rly planted beet the pine forest climbed over gently undulating hills. 'Yo
61. ked up at the dim stairway which climbed steeply out of the bare and musty

****

And, metaphorically ...

62. for first-time buyers trying to climb on to the first rung of the housing

****

With 'down'

63. tion of running water, attempt to climb down the slippery cemented sides of
64. t an ice axe he would be lucky to climb down fifty feet without falling. It
65. third floor but people there had climbed down from the balconies and were

****

[PLAYER] climb above [PLAYER]

A cliche. Genre: British sports journalism

66. w-in enabling Chris Fairclough to climb above defenders and head past Carte
67. rom their second corner, Robinson climbed above static defenders to head Ga
68. er 38 minutes when Alan Kernaghan climbed high to Putney's corner and heade

---

fig. 3 cont.

IV.

[ABSTRACT] climb ([AMOUNT]) [ADVERBIAL-AMOUNT]

69. uring wage costs will accordingly climb by 4 per cent in 1990 and wages in
70. the good: coal prices look set to climb by 80 per cent over the next 25 yea
71. a week of losses ended as the MIB climbed 10 points to 1,088, boosted by fo
72. ined 6p to 227p and Racal Telecom climbed 12p to 342p. STC was the subject

****

NP climb [AMOUNT] PP

73. to raise money for diabetic children &dash. by climbing 15,000 feet up Moun
t Kilimanjaro. Siste
74. The road angled towards the rim of the valley, climbing 2,000 feet in eight
relentless miles. T

fig. 3 cont.

Some Definitions

The following definitions summarize some of the points made in this paper. They do not necessarily reflect the common acceptation of the words being defined.

Activated beliefs: a hearer's beliefs about the meaning of the words used by an utterer for some communicative purpose; an utterer's beliefs about (and reasons for using) those words.

Meaning potential: the potential that a word has to contribute to the fulfilment of an utterer's communicative purpose.

Text: a sequence of words designed to activate beliefs (turn meaning potentials into meanings).

Dictionary definitions: a list of the meaning potentials of lexical items.

Corpus: a collection of texts.

Corpus technology: facilities to analyse computationally the patterns of word use found in a corpus, thus providing a syntactic framework onto which meaning potentials may be projected.

Semantic type: a more delicate level of syntactic classes than traditional part-of-speech classes. Together with *syntactic patterns*, they enable us to show how meaning potentials are realized as meanings.

Convention: any of the recurring patterns of word use found in a corpus, which are associated with meaning potentials.

Exploitation: the way in which a convention is used. Linguistic conventions are exploited in just the same way as the maxims described by Grice (1975) are exploited: they may be fulfilled, or they may be flouted.

Flouting: the mechanism by which literary style, metaphors, and meaning changes take place. Only part of a convention may be flouted in any one use. Too much flouting results in a communication breakdown, as when Lewis Carroll's Humpty Dumpty used *glory* to mean 'a nice knock-down argument'.

Contrastive Classes - Relating Monolingual Dictionaries to Build an MT Dictionary

ULRICH HEID

Abstract

In lexicalistic grammatical theories, ways of hierarchically structuring monolingual lexicons have been proposed. We show in this paper[1] how such devices for lexical organization (classes and instances, a class hierarchy) can also be used to structure contrastive dictionaries for machine translation.

Within a relational modeling of a transfer based approach to MT, we organize the monolingual dictionaries of verbal subcategorization in hierarchies and then relate such monolingual classifications to form a contrastive one. The resulting contrastive classes are not only a useful device to avoid redundancy in the lexicon, they also allow to express generalizations at the level of lexical translation problems.

The first section of the paper outlines the state of the art in contrastive classifications, in MT research and in lexicography and summarizes the criteria we use in our own contrastive classification. Section 2 describes devices for the structuring of monolingual and contrastive dictionaries. Section 3 is about the integration of contrastive classes into an MT dictionary and serves to describe the framework and the devices used, as well as to discuss an example.

[1]This work has been carried out in part in the framework of the DELIS project (LRE 61.034) on descriptive lexical specifications and corpus based lexicon building. DELIS is financed in part by the European Commission under the Linguistic Research and Engineering (LRE) programme of its Directorate General XIII E (Luxembourg). Part of the work has been carried out as well in the VERBMOBIL project, funded by the German BMFT.

The author would like to thank Andreas HAIDA, Martin EMELE and Stefan MOMMA for their comments on an earlier version of this paper. Andreas HAIDA has integrated contrastive classes into an existing modeling. All misconceptions and errors are of the author's responsibility.

1 Types of contrastive problems in the lexicon

The idea that lexical contrasts between languages can be classified is not at all new. Such tentatives have been carried out, in an informal way, in contrastive stylistics, in translation science and in translators' training. Typical examples are the books by [Malblanc 1968] and [Vinay/Darbelnet 1958]. These books are collections of problems translators often encounter. The classifications operated there are not based on formal criteria, and not related to any formal representation of lexical material. The approach followed could be described as onomasiological: the authors note that a given content is rendered by different linguistic means in the languages contrasted.

In machine translation and computational lexicography, the use of classifications of contrastive lexical problems has much less tradition, however. Certainly, work like TALMY's on the linguistic description of movement expressions can be thought of as establishing classes of contrasts between e.g. English and French; but the use of contrastive classifications in machine translation systems or in their dictionaries has only recently been perceived as an advantage. The lack of means to make use of lexical classification has been pointed out, for example, for the METAL system, by [Fontenelle/Adriaens/De Braekeleer 1992].

1.1 Problem classfications in MT

The few tentatives to make use of a classification of contrastive problems, in machine translation (MT), are mostly characterized by one or more of the following tendencies:

- partiality of classifications: only the fragment treated by a given (module of a) machine translation system is covered (see e.g. work by [Bemova et al. 1988] (types of noun phrases and verb phrases), etc.); the classification is then not generalizable;
- system-bound classifications: the description of contrastive problems is made only or predominantly in terms of the representations used by a given machine translation system (e.g. EUROTORA structures in [Lindop/Tsujii 1991]'s work), and in terms of operations performed on these representations (cf. [Thurmair 1990]'s discussion of tree operations in METAL: delete, add, modify (sub-)trees); of course, the representation of lexical phenomena underlying the system and its dictionary is important for the formulation of contrastive classes, but the problem, with the classifications mentioned here, is that they can not easily be applied outside the systems for which they have been developed, because, for example, in unification-based systems, some of the classes they establish would be trivially treated by the grammars, and would thus not constitute a problem at all;
- non-strict classifications: examples are classified into more than one class; this makes it difficult to derive systematic rules for the actual treatment of contrastive problems (this problem has been pointed out by [Vandooren 1993].).

Recent work in MT has led to more general proposals for problem typologies. DORR (cf. [Dorr 1990], [Dorr 1992]) has established a classification of what she calls 'syntactic' and 'lexical-semantic divergences'; she uses these classifications as a basis for the formulation of rules and parameter settings in a machine translation system the grammars of which are inspired by the "principles and parameters" approach.

One of DORR's main motivations is also true for our own work on an MT dictionary: if we manage to *classify* some of the problems we have to treat in the contrastive lexicon, we can *treat similar problems with similar techniques*, and organize contrastive dictionaries according to the types of operations needed to cater for the respective types of contrastive problems.

DORR concentrated on cases where no or few semantic changes occur between source language sentences and their target language equivalents (she calls these cases "divergences"). [Kameyama/Ochitani/Peters 1991] describe cases where one language has more specific lexical material than another, i.e. where one of the two languages makes more distinctions, lexical semantic or other, than the other language, and where the "poorer" language does not have lexical means to express the distinctions made in the "richer" one. They call these cases "mismatches".

With [Kameyama/Ochitani/Peters 1991], divergences can be seen as a special case of mismatches: in divergence cases, the differences of source and target language with respect to a given descriptive dimension can be "egalized" by lexical (and/or grammatical) means within a sentence. Necessarily, then, changes at the level of syntactic construction and possibly of the syntax-semantics interface or of the "distribution" of meaning components over the lexical material (*swim across the river* ↔ *traverser le fleuve à la nage*) occur. "True mismatches" are those where no lexical means are available in the target language to express distinctions made in the source language.

1.2 Problem classifications in lexicography

Traditional bilingual lexicography has paid much attention to equivalence gaps and has in particular identified lexical semantic and cultural reasons for these (cf. e.g. work by [Kromann 1987] on this topic from the point of view of directional dictionaries). Equivalence gaps can be seen as subtypes of "mismatches".

In the recently completed work of the MULTILEX project, a contrastive lexical problem classification has been proposed with a view to defining the types of information needed in dictionaries which would support, among other applications, a transfer-based MT system. Classes established there include "hypernymic" and "hyponymic" translation, as well as "variant" and "related" translation. The terminology used is different from that of machine translation research, but a closer analysis of the definitions of the classes and of the examples discussed shows a great deal of overlap.

The table in Fig. 1 contains a very rough comparison of the broad classes used by DORR, by [Barnett/Mani/Rich 1992], who have summarized the discussion about contrastive lexical problems in MT research, and by the MULTILEX project[2].

1.3 Combining the classifications for practical purposes

Work in translation lexicography and in machine translation emphasizes different aspects of contrastive problems. But by combining the relevant distinctions made in previous work, we arrive at a quite stable classification which we can use for our purposes.

[2]The class of *variant translation* in MULTILEX is one where source and target language only have differences at the stylistic or connotational level ("stylistic divergence"). This type of translation problems is not described (n.d.) in the MT research we analyzed.

MULTILEX	BARNETT	DORR
complete equivalence – + transformations within one sentence	divergence	divergence
variants (e.g. of style, connotation, etc.)	n.d.	n.d.
partial – hypernymic – hyponymic – related	mismatch upward move downward move sideward move overlap	(mismatch)

Figure 1: Classifications of contrastive lexical problems

The toplevel distinction is that between mismatches (in the sense of [Kameyama/Ochitani/Peters 1991]) and divergences (i.e. mismatch problems which can be egalized by lexical or grammatical means).

Mismatches can involve

- a "loss of information", which is equivalent to "hypernymic translation" (in MULTILEX) or to an "upward move" in the concept hierarchy used by e.g. [Barnett/Mani/Rich 1992];
- a "need for overspecification" ("hyponymic translation", "downward move");
- an "overlap" in the information content of source and target language ("related translation", "sideward move").

It should be noted that, the same way as the lack of descriptive dimensions does (in the case of mismatches), also other parameters can lead to the need for "divergence-type" translation; for example, the theme/rheme organization of sentences in a text may require changes in the realization of target language sentences, even if there is a structurally isomorphic and semantically adequate equivalent.

Divergences can be further classified according to the type of linguistic object(s) the realization of which differs between source and target language, and according to the levels of description involved[3].

- The lexeme translated is differently realized itself:
  - categorial divergence: EN *be hungry* ↔ FR *avoir faim*;
  - differences of syntactic realization of the lemma in question (DE *das besteht aus zwei Teilen* ↔ EN *this is composed of two parts* (active/passive, cf. [Thurmair 1990])).
- The syntactic environment of the lexeme to be translated is different:
  - differences in the syntactic realization of complements subcategorized by the lexeme to be translated;

[3]In the following, we use DORR's terminology whenever appropriate (DORR's terms for the contrastive classes are then between quotes).

  - "conflational" divergences: cases of incorporation of arguments or modifiers (for instance EN *to mispronounce* ↔ FR *prononcer de travers*).

- The mapping between syntax and semantics, i.e. the "linking rules" describing the relationship between arguments and complements, is different in source and target language: "thematic" divergences (EN *I miss my dictionary* ↔ FR *mon dictionnaire me manque*).

- Both, properties of the lexeme translated and of its syntactic environment are concerned in cases of *head-switching* (EN *he still plays piano* ↔ FR *il continue à jouer du piano*), in cases where verbs of one language are translated by support verb constructions of another, etc.

It should be noted that the types can cooccur and thereby further constrain the set of possible equivalents; it is thus important to be able to separate out individual cases, and to treat them by use of individual "rules". If we have a means to treat individual classes of contrastive problems separately, the individual rules can be combined to cater for more complex cases.

2 Lexical classes in monolingual and bilingual MT dictionaries

The basic assumption of our work is that classification devices used in the formal modeling of monolingual dictionaries can as well be put to profit in the construction of bilingual dictionaries.

One of the interesting principles of HPSG (cf. [Pollard/Sag 1987], [Pollard/Sag in press]) is the use of a hierarchical classification for the construction of monolingual subcategorization dictionaries. HPSG represents linguistic objects by typed feature structures, which can be organized in subsumption hierarchies, to capture generalizations.

Other unification grammars only have less powerful devices for structuring the monolingual dictionary; an example are templates in Lexical Functional Grammar (LFG, cf. [Bresnan 1982]): a "template" is defined (like a macro in programming languages) and stands for a certain subcategorization pattern; it contains a variable at the place of the verbal predicate of the LFG-typical predicate-argument structures. So, the descriptions of a transitive reading of FR *acheter* and of an intransitive one of *venir*, which look like (1) and (2) in the table in (Fig. 2), can be produced by application of the definitions of the transitive, (3), or intransitive, (4) template, and their application to the verbal predicates, as indicated in (5) and (6) of (Fig. 2).

By using templates, we only need to add statements like those in (5) and (6) of (Fig. 2), each time an additional transitive or intransitive verb is added to the subcategorization dictionary.

This functionality can be achieved with types as well. But, on top of this, we can make use of and of inheritance of properties (e.g. of passivization); moreover, if we combine two monolingual subcategorization dictionaries, the types (and the type checking mechanisms inherent to typed feature logic based formalisms) can be used to constrain equivalence

No.	template definition	verb entry	subcategorization frame
(1)		acheter, V	(↑ PRED) = "acheter <(↑ SUBJ) (↑ OBJ)>"
(2)		venir, V	(↑ PRED) = "venir <(↑ SUBJ)>"
(3)	@transitive (x):-	x, V,	(↑ PRED) = "x <(↑ SUBJ) (↑ OBJ)>"
(4)	@intransitive (x):-	x, V,	(↑ PRED) = "x <(↑ SUBJ)>"
(5)		@transitive (acheter)	
(6)		@intransitive (venir)	

Figure 2: Simple examples of subcategorization templates for LFG

descriptions in a contrastive classification. Monolingual syntactic hierarchies, e.g. for verb subcategorization, have been developed, for example, in [Sanfilippo 1993].

Our goal in introducing contrastive classes is to relate two monolingual subcategorization dictionaries: suppose we have hierarchical monolingual syntactic classifications of the above type, for source and target language; by relating these, we have, for example, a compact way of stating that a German transitive verb is translated by a French transitive verb.

The questions which need to be answered in this context are the following: if such a compact way of describing interrelationships between syntactic properties of source and target language equivalents is available, what is the status of such device? It is just an abbreviation which can be used in an HPSG-like system, to make dictionaries less redundant and dictionary updates more efficient, or is there also a way to use the device to account for contrastively relevant cases, as they have been described in the above problem classification?

3 Integrating contrastive classes into a transfer dictionary

3.1 Framework and Modeling

Our framework is a transfer-based approach with a relational modeling of equivalence, as proposed by [Zajac 1989][4]. The prototyping is done in the typed feature structure based TFS system (cf. [Emele 1993], [Zajac 1992]), which provides an inference machine for typed feature structures.

The major relevant properties of our prototypical modeling are as follows: We use functional structures (f-structures) of LFG as the level of representation on which we formulate transfer correspondences (i.e. equivalence statements). This is taken over from [Zajac 1989] and [Zajac 1992]; the basic mechanisms of this approach to translation go back to [Kaplan et al. 1989].

In the following examples, we only indicate predicate-argument structures, which are the part of LFG dictionary entries primarily involved in the correspondence statements. Other than in standard LFG, where the phrase type of each constituent is annotated in the c-structure and thus need not be explicit in the f-structure, we follow ZAJAC's approach of annotating the phrase structural construct of each constituent as a type of the feature

[4]See, however, below, section 4 and the work described in [Kuhn/Heid 1994], for extensions.

structures representing the complements of a verb. An entry like that in (1) of (Fig. 3) is thus rewritten in our TFS notation[5] as in (2):

(1) déconseiller, V, (↑ PRED) = <(↑ SUBJ) (↑ OBJ) (↑ OBJ2) >

(2) f-vp [PRED: *"déconseiller"*,
SUBJ: f-np,
OBJ: f-np,
OBJ2: f-pp [PREP: "*à*"]].

Figure 3: An LFG predicate-argument structure rewritten according to [Zajac 1992]'s notation

More precisely, (2) is a partial structure for a verb phrase of the French grammar and lexicon specification (type f-vp), which has a PRED(icate), a SUBJ(ect), an OBJ(ect) and an indirect OBJ(ect)2.

Translation equivalence is modeled as a reified relation. Thus we have only one data structure, namely (typed) feature structures, for both monolingual descriptions and equivalence relations[6]. Reified transfer relations (which we call *transfer statements*) are feature structures with two attributes[7], one for each of the contrasted languages (e.g. FF, FE, FD in the examples below); the values of these attributes are LFG f-structures of the monolingual grammars and lexicon entries. In the transfer statements, the f-structures may be only partially specified.

TFS supports monotonic specialization hierarchies; our modeling of lexical classes in monolingual dictionaries (cf. the discussion of LFG's templates, above) and of contrastive classes makes use of this property: a lexical class is a type which is underspecified with respect to the value of the PRED attribute, or no (or only a generic) value for it. Individual contrastive lexical entries are subtypes or instances of this type: entries are more specific than classes insofar as they contain concrete values for the "PRED" attribute and, possibly, additional constraints.

It is possible to add constraints to the contrastive description. These concern the type of the values which can appear under an attribute, a particular value of an attribute (e.g., ind the case of allocations, to express that a given equivalence statement for a verb only holds if a certain lemma occurs as subject or object of this verb), or an additional *condition* of arbitrary complexity. Conditions are used, for example, to describe the equivalence relation for arguments of a verb. Intuitively, we want to express that the equivalence relations between two verbal predicates, say FR *déconseiller* and DE *abraten*, only holds if, for a pair of sentences, also equivalence relations between the subcategorized complements of *déconseiller* and of *abraten* exist.

So for example, the SUBJect of *déconseiller* and the SUBJect of *abraten* must be related by an equivalence relation, an so on for all of the complements. We indicate, in (Fig. 4), the

[5]Notation: Attribute names are in SMALL CAPITALS, type names in sans-serif letters; GRAMMATICAL FUNCTIONS are notes as ATTRIBUTES, whereas the partial structures we handle are typed according to their phrase type.

[6]This makes a difference with those MT systems which have an extra data type to express relations or mappings, between levels of monolingual description or between linguistic objects from two languages.

[7]We are aware that the "language-attribute" is an element of a contrastive metadescription. So are the "prefixed" type names of phrasal categories (e.g. f-np, f-vp, e-vp): to avoid these, other technical solutions could be found, such as separate namespaces for type names, etc. We leave them here for the sake of expository clarity.

predicate-argument structure for DE *abraten*, and we then reproduce, the bilingual FR ↔ DE entry for the equivalence pair; we reproduce the notation used in the TFS statements compiled by the TFS system[8]:

```
d-vp [PRED:     "abraten",
      SUBJ:     d-np [CASE: nom],
      OBJ-2:    d-np [CASE: dat],
      von-OBJ:  d-pp [PREP: "von"]].

tr-017 [FF: f-vp[PRED:    "deconseiller",
                 SUBJ:    #f-subj,
                 OBJ:     #f-obj,
                 OBJ-2:   #f-obj2],
        FA: d-vp[PRED:    "abraten",
                 SUBJ:    #d-subj
                 OBJ-2:   #d-obj2,
                 von-OBJ: #d-von]]
                                    :-tr[FF: #f-subj,
                                         FA: #a-subj,],
                                      tr[FF: #f-obj,
                                         FA: #a-von],
                                      tr[FF: #f-obj2,
                                         FA: #a-obj2].   tr-017 < tr.
```

Figure 4: German entry for *abraten* and transfer entry for *abraten* ↔ *déconseiller*

The values of the grammatical function attributes are coindexed[9] with the respective transfer statements in the conditions, which describe the relationship of the complements of the source and target language lexemes.

The example of *abraten* ↔ *déconseiller* is an instance of a "thematic" divergence, as illustrated by the following examples. It is similar to the cases much discussed in the MT literature (FR *manquer*/EN *miss*, EN *like*/SP *gustar*, etc.).

DE *Die Ärzte raten Schwangeren von diesem Medikament ab.*
FR *Les médecins dé conseillent ce médicament aux femmes enceintes.*

3.2 A complex example

Cases of head switching are another contrastive class which has received much interest in the MT literature: DE: *er schwimmt gerne* ↔ FR: *il aime nager*. This class (discussed among others by [Kaplan et al. 1989], [Zajac 1989], etc.) has many elements in German/French translation. A few examples are given below:

- anscheinend, offenbar ⟷ sembler INF

[8] In (Fig. 4), we have three conjunctively related conditions, one for the mapping of each complement. Notation: conjunction is expressed by the comma operator (","), disjunction by "|". The conditions are introduced by the sign ":-", all statements end with a period (".").

[9] Notation: A text preceded by the "#" sign is a coindexing marker. Coindexed elements are interpreted as token-identical. The variables serving as coindexing markers are local to the statement they appear in.

- unerwartet ⟷ venir à INF
- schließlich ⟷ finir par INF
- allmählich (...werden) ⟷ commencer à INF
- immer noch ⟷ continuer de/ à INF
- sogar ⟷ aller jusqu'à INF
- unbedingt ⟷ tenir à INF

We use the following pair of example sentences to illustrate our modeling:

DE *Er spielt immer noch Fussball.*
FR *Il continue à jouer au football.*

In some cases, German aspect marking adverbs or other sentential adverbs are translated into French by constructions with a verb which takes an infinitival (or sentential) complement. The contrastive class (cases with infinitive in French) is described by the following statement, in (Fig. 5)[10]:

```
tau-advv [FD: d-advp [ARG: #D-Arg],
          FF: f-vp [XCOMP: #F-XComp]]   :-  tau [FD: #D-Arg,
                                                 FF: #F-XComp].
```

Figure 5: Equivalence statement for the translation of sentential adverbs of German by French verbal constructions

The lexical entry for e.g. DE *immer noch* ↔ FR *continuer à* relates the two lexemes; it is a subtype of the `tau-advv` relation; the syntactic schema specified in `tau-advv` is thus the only one which can be used to analyze and translate sentences with *(immer) noch* and *continuer à*.

3.3 Discussion

The modeling of contrastive classes according to the approach sketched here has a number of advantages, but also some limitations.

We have gained experience, so far, in the modeling of examples illustrating the main classes of divergences. Also, certain types of mismatches of the hypernymic/hyponymic kind can be captured in contrastive generalizations, when the representation used for transfer includes a hierarchy of partial lexical semantic representations shared by source and target language.

A major limitation is the partiality of the approach: not the entire vocabulary of a machine translation system can be organized in contrastive classes. However, in comparison with the problems of redundancy encountered in systems which do not offer this possibility (see e.g. [Fontenelle/Adriaens/De Braekeleer 1992]'s discussion of "grooming verbs" in METAL), any

[10]These cases have been treated in detail by Andreas HAIDA, who has integrated them into ZAJAC's system, at IMS-CL, Stuttgart.

partial solution is an advantage, although it may be seen rather as an engineering advantage than as a conceptual one[11].

Among the advantages of the proposed approach, the following seem most important to us:

- redundancy in contrastive dictionaries can be reduced; thereby the dictionaries become easier to maintain and to extend; we are aware that this advantage is more practical (and technical) than conceptual in nature; but for practical usability of dictionaries for MT systems, this may nevertheless be important;
- contrastive classes can be specified "locally": there is no need to include information about embedded structures, if this does not play a role for the description of the actual contrastive problem; thereby contrastive statements become more independent from each other and thus more freely combinable. Consider, for example, the translation of DE *Er telephonierte zufällig* by EN *He happened to be on the phone*: the translation of DE *zufällig* by EN *happen to* (head switching) follows the principles of our statement in (Fig. 5), above. This statement is completely independent and thus freely combinable with a statement for the translation of DE *telephonieren* by EN *be on the phone*. Thus, we can combine translation statements without having to bother about the interaction of the statements. Given that they relate only well-formed structures of source and target language, the monolingual grammars strict their combinability.

An interesting point for further discussion concerns the status of the contrastive classes. It is evident that certain pariring of subcategorization properties of source and target language are mainly a matter of "contrastive hazard" (what is typical about "*jemandem danken* ↔ *remercier qn*"?). Other such classes, as shown in the example of the aspect marking adverbials vs. verbal periphrases, seem to be semantically somehow coherent. This is what happens, for example with the subclasses of verbs of motion in French, compared to German or English (cf. *er schwimmt in das Loch (hinein)* ↔ *il entre dans le trou en nageant*, etc.).

4 Summary, future research

We have sketched an approach to contrastive lexical modeling which takes inspiration from contrastive problem classifications from MT work and lexicography. Divergences and hyponymic/hypernymic mismatches can to a large extent be modeled as contrastive lexical classes which abstract away from actual lexemes; such statements relate partial syntactic structures (in our case f-structures of LFG) from source and target language, but are used as a class definition; individual lexical entries are instances of such classes, and they add only the equivalent PRED(icates), whereas all other information is inherited from the class definition. We discuss a few examples taken from a modeling in the Typed Feature Structure System, TFS.

Here, we use an approach to translation which operates a transfer at the level of predicate-argument structures. In [Kuhn/Heid 1994], we have used DORR's classification of contrastive problems to verify whether and how an interlingua-based approach can cater for

[11] [Nirenburg/Levin 1991] give arguments against an approach to machine translation which would exclusively use classes and operations such as those described here. They favour the use of much deeper representations as just predicate-argument structures (op. cit., pp. 12-14).

the problems described here. To that end, we have related small grammar fragments and lexicons of HPSG, for French and English, and we have described the mappings which occur at the syntax/semantics interface in the HPSG lexicon. These mappings could in principle again be modeled as classes, with individual pairs of lexical entries of equivalents as instances[12]. These experiences suggest that the phenomena described here can in principle be described as contrastive classes, independently from the approach to MT-directed modeling (transfer or interlingua), as long as the nature of the mappings (either at the syntax/semantics interface, in the case of an interlingua approach, or at the transfer level) is precisely defined.

References

[ACL-29 1991] Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, University of California, Berkeley, California, USA, 1991,

[Barnett/Mani/Rich 1992] James Barnett, Inderjeet Mani and Elaine Rich: "Reversible Machine Translation: What to do when the Languages don't match up", in: Strzalkowski, T. (ed.): Reversible Grammar in Natural Language Processing. Boston/Dordrecht/London, 1994.

[Bemova et al. 1988] Alevtina Bemova, Karel Oliva and Jarmila Panevova: "Some Problems of Machine Translation Between Closely Related Languages", in: Prodeedings of COLING 1988.

[Bouillon/Clas 1993] Pierette Bouillon and André Clas (Eds.): *Etudes et recherches en traductique. Problèmes de traduction par ordinateur.* Montréal 1993.

[Bresnan 1982] Joan Bresnan: *The Mental Representation of Grammatical Relations*, (Cambridge, Mass.: The MIT Press) 1982.

[Dorr 1990] Bonnie Dorr: "Solving Thematic Divergences in Machine Translation", in: *Proceedings of the 28th Annual Conference of the Association for Computational Linguistics.* (Pittsburgh, Pa.: University of Pittsburgh) 1990, pp.127-134.

[Dorr 1992] Bonnie Dorr: "Interlingual Machine Translation: A Parameterized Approach". Draft for AI Journal 1993, 1992.

[Emele 1993] Martin Emele: "TFS – The Typed Feature Structure Representation Formalism". Manuscript, Kuhn, J. and Heid, U.: Treating sturctural differences in an HPSG-based approach to interlingual machine translation. Extended abstract for the Jahrestagung der DGfS 1994, AG-6.

[Fontenelle/Adriaens/De Braekeleer 1992] Thierry Fontenelle, Geert Adriaens and Gert De Braekeleer: "L'unité lexicale dans le système de traduction assistée par ordinateur METAL", in: ce volume, 1992.

[Kameyama/Ochitani/Peters 1991] Megumi Kameyama, Ryo Ochitani and Stanley Peters: "Resolving Translation Mismatches With Information Flow", in: [ACL-29 1991], 1991.

[Kaplan et al. 1989] Ronald M. Kaplan, Klaus Netter, Jürgen Wedekind and Annie Zaenen. "Translation by Structural Correspondences" In: *Proceedings of the Fourth Conference of ACL, European Chapter*, Manchester, 10-12 April 1989.

[12] If modeled in a relational system like TFS, the relations used to express the syntax-semantics mapping, and those used to model transfer relations, as shown in this paper, are formally of the same nature.

[Kromann 1987] Hans-Peder Kromann: "Neue Orientierung der zweisprachigen Wörterbücher. Zur funktionalen zweisprachigen Lexikographie", in: Snell-Hornby, M. and Pöhl, E. (eds.): Translation and Lexicography. Papers read at the EURALEX Colloquium held at Innsbruck 2-5 July, 1987.

[Kuhn/Heid 1994] Jonas Kuhn and Ulrich Heid: "Treating structural differences in an HPSG-based approach to interlingual machine translation", to appear in *Proceedings of the Jahrestagung der Deutschen Gesellschaft für Sprachwissenschaft 1994*, 1994, ms. Stuttgart, 26pp.

[Lindop/Tsujii 1991] Jeremy Lindop and Jun-ichi Tsujii: "Complex Transfer in MT: A Survey of Examples", manuscript (Manchester: UMIST, Center for Computational Linguistics), no.91/5.

[Malblanc 1968] Alfred Malblanc: "Stylistique comparée du français et de l'allemand". Paris, 1968.

[Nirenburg/Levin 1991] Sergei Nirenburg and Lori Levin: "Syntax-Driven and Ontology-Driven Lexical Semantics", in: Pustejovsky, J. and Bergler, S. (eds.): Lexical Semantics and Knowledge Representation. First SIGLEX Workshop, Berkeley, CA, USA, June 17, 1991.

[Pollard/Sag 1987] Carl Pollard and Ivan Sag: *Information based Syntax and Semantics.* CSLI Lecture Notes, Vol.13, (Chicago: Chicago University Press) 1987.

[Pollard/Sag in press] Carl Pollard and Ivan Sag: *Head Driven Phrase Structure Grammar*, University of Chicago Press and CSLI publications, in press

[Sanfilippo 1993] Antonio Sanfilippo: "LKB Encoding of Lexical Knowledge", in: Briscoe, T., Copestake, A. and de Paiva, V. (eds.): Inheritance, Defaults and the Lexicon. Cambridge, 1993.

[Thurmair 1990] Gregor Thurmair: "Complex Lexical Transfer in METAL", in: [TMIMT-3, 1991], pp.91-107.

[TMIMT-3, 1991] *Proceedings of the 3rd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language, 11-13 June 1990.* (Austin: University of Texas) 1990.

[Vandooren 1993] Françoise Vandooren: "Un exemple de problèmes syntaxique: les divergences de traduction entrel l'Anglais et le Français", in: [Bouillon/Clas 1993]

[Vinay/Darbelnet 1958] J.-P. Vinay and J. Darbelnet:"Stylistique comparée du français et de l'anglais. Méthode de traduction, in: [Malblanc 1968]

[Zajac 1989] Rémi Zajac. "A transfer model using a Typed Feature Structure rewriting system with inheritance", in: *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver, 1989.

[Zajac 1992] Rémi Zajac: "Inheritance and Constraint-Based Grammar Formalisms", in: *Computational Linguistics, 18 (2)*, [= Special Issue on Inheritance: I], 1992, pp.205-218.

A Dictionary for Language Generation

ADAM KILGARRIFF

Abstract

The Longman Language Activator is a dictionary designed to help learners of English generate fluent, natural English. A fully marked-up electronic version has now been prepared. We discuss its structure and theoretical underpinnings, and consider its potential for natural language processing in general and generation in particular.

1 Introduction

From a review of the literature on machine-readable dictionaries, one might well conclude that dictionary publishers are not human. That strange beast, the dictionary, is tackled as if it had arrived from outer space: there must be some structure and reason to it, amongst the font-change codes and unprintable characters, and, by jingo, we'll stare at it until we find it! While this attitude is in the finest tradition of thoroughbred empiricism, and is sometimes the only course available, there is another option: ask.

Longman have produced an electronic version of the Longman Language Activator (?) in which such questions are answered even before being asked. It is in SGML, and all text elements are marked up with tags to show what type of information they carry; whether it is headword, part of speech, non-standard inflection, variant spelling and so on. This is now available for NLP research. In the same way that foreign learners' reference dictionaries have been particularly useful for natural language understanding, we anticipate that this dictionary - a generation dictionary, the first of its kind - will prove particularly useful for natural language generation.

The Activator is based on Longman's spoken and written corpora, and is the product of a five year research project. In the course of that time, the core lexicon of English has been analysed to give a framework of 1000 concepts, and each of the 20,000 words and phrases in the core lexicon is located under one of these concepts. Unlike a thesaurus, each of these lexicalisations is defined, with examples given, as in a traditional dictionary. In these definitions, particular care is taken to ensure that the user will be able to identify the appropriate word for what they want to express - whether, for example, it is *stare*, *gaze* or *gape*. The definitions use the Longman defining vocabulary, as used in LDOCE (?, ?), and for each definition, one or more example is given.

In Section 2 we look more closely at what the Activator is. In Section 3 we consider how the conceptual framework was developed, and contrast it with other approaches. In Section 4 we consider what its taxonomy offers to AI and NLP and in Section 5 we consider its potential for the problem of lexical choice in natural language generation.

2 The Activator

2.1 Rationale

A dictionary could and should provide enough information for the *generation* of appropriate and stylistically correct English. This has been the guiding principle of the Activator project. English language teachers and students repeatedly say that current ELT dictionaries fail to help them with the matter of "choosing the right word", that is, knowing when one word rather than another is suitable for the context. It was as a response to this lament that the idea of the generation dictionary was born.

For human learners of English and for some varieties of NLG system, a crucial step in the generation of natural English is the move from a general term carrying the core of the message to be expressed, to a specific word or phrase. The specific word or phrase will be selected from a group of near-synonyms and must be chosen with due regard for its 'meaning' as most widely construed, that is, including: evaluative content;

levels of formality and emphasis; any connotations, implications or presuppositions; and the subjects, objects, modifiers and other collocates with which the word sounds most natural. It is this step of the generation process that the Activator addresses, and to this end, it is organised taxonomically, to guide the user from the general concept to the specific lexicalisation.

2.2 Description

The Activator taxonomy has three levels: concept, section, and lexicalisation. A concept is divided into sections, each of which has a definition, and the possible lexicalisations are listed and defined under one of the sections. Thus, taking a run of the book at random, concepts include

START DOING STH	START STH/MAKE STH START
START TO HAPPEN, EXIST ETC	STAY/NOT LEAVE
STAY WITH SB, IN A HOTEL ETC	STEAL
STICK	STICK OUT

and STILL.

Under STEAL, there are nine sections, each of which is listed with its possible lexicalisations below.

STEAL

1 to take something that does not belong to you

steal	**go off with/**	**take**	**rip off**
pinch	**walk off with**	**help yourself to**	**nick**

2 to steal something that is not very valuable or important

pilfer	**snitch**	**swipe**

3 to steal things from a house, shop, bank etc

burgle	**loot**	**burglarize**	**hold up**
rob	**knock over**	**shoplift**	

4 to steal something from a person, particularly in a public place

rob	**mug**	**snatch**

5 to steal money that you have been trusted to look after, especially from the place where you work

embezzle	**misappropriate**	**have your fingers in the till**

6 someone who steals

thief	**shoplifter**	**robber**	**mugger**
burglar	**kleptomaniac**		

7 the crime of stealing

theft	**larceny**	**burglary**	**shoplifting**
robbery			

8 a particular act of stealing

theft	**hold-up**	**burglary**	**stick-up**
robbery	**raid**	**break-in**	**job**

9 words for describing something that has been stolen

stolen	**loot**	**haul**

The user either looks up STEAL, and gets directly to the concept, or *rip off*, *rob* etc. and is redirected from an index entry to the STEAL entry. They then look at the definitions for the various sections, and determine which fits the idea they wish to express.

Perhaps the user wants to talk about stealing from a shop: they then move to section 3.

3 to steal things from a house, shop, bank etc

burgle to illegally enter a building, especially a house or office, and steal things from it [*v* T]
burgle sth *It would have been easy to burgle the Davis' house because they always left the bedroom window open ...*| **have your house/flat/place etc burgled** *Any employee who has had his car stolen or his house burgled will need time off work.*

burglarize an American word meaning to illegally enter a building, especially a house or office, and steal things from it [*v* T]
We had never been burglarized in broad daylight before ...

rob to steal money or other property from a bank, shop, or other public building [*v* T]
Two men robbed the Central Bank yesterday, escaping with one million dollars ...

shoplift to steal things from shops by taking them from shelves and hiding them under clothes or in a bag [*v* I]
The boy was caught shoplifting with several cassettes in his pocket. ...

⋮

The user looks at the lexicalisations listed under that section, and identifies that, for example, if the event involved stealing by taking things from shops and hiding them under clothes, the appropriate word is *shoplift.*

2.3 Inclusion

For a reference dictionary, the answer to the question "How many words should it contain?" is "as many as possible". There is no knowing what words the user will encounter in the texts he, she or it hears or reads, and it is the job of the lexicographer to arm him, her or it for as many eventualities as possible. For a generation dictionary, the situation is different. The dictionary need only be sufficient to allow the expression of what the user might want to express. The Activator is targeted at slightly more advanced students than LDOCE yet the headword list, at 20,000, is around one third as long. An active vocabulary need not be as big as a passive vocabulary, but, if words are to be used correctly, much more must be known about each item. The Activator is a fatter book than LDOCE.

For reference dictionaries, particularly when used with written material, the word as delimited by spaces and punctuation is the key. [1] It is the item that can be looked up. Often the space-delimited word is not the linguistically salient unit, and modern dictionaries work hard at the difficulties this presents, but it must still play a prominent role. By contrast, the key in a generation dictionary is the concept, so the lexicographer is free to include phrases on a par with single words as lexicalisations. Our experience of language corpora show us the extent to which natural English is built from phrases and other multi-word expressions. This is often particularly vivid when looking at the spoken corpus, where we find, for example, that a common lexicalisation for OBVIOUS, with the meaning is the expression **you only have to** ..., as in "You only have to look around you to see how many families have two cars." In the Activator a very high proportion of lexicalisations are phrasal.

A third contrast is less with reference dictionaries than with other classification schemes. These have tended to concentrate on objects, and the common nouns that denote them. In the Activator the emphasis is on that part of the core vocabulary where learners have most difficulty choosing the right word. This is not the real world vocabulary of tables and chairs, where there is likely to be a straightforward correspondence between, e.g., *armchair* in English and *fauteuil* in French, and a bilingual dictionary provides straightforward answers. It is the more abstract, more often verbal or adjectival, more often phrasal territory. In most dictionaries, nouns account for over half the headword list. For the Activator, the figure is 25%.

2.4 Lexicographic style

The Activator was a large project with thirty authors and editors over five years, and to maintain consistency over the project it was necessary to define clear policies stating how different items of information wre to be expressed. This is a point well worth making in the world of machine-readable dictionaries. The definition of *burglarize* starts

> an American word meaning ...

This is not an aberrant definition, inevitably doomed to scupper attempts to automatically identify genus words. Rather, it is one of a short list of set phrases used to introduce

[1] Particularly in a language such as English which has little prefixation.

definitions where the word or phrase is marked in some way. It was decided that information about markedness was more accessible to the student if presented in English rather than in a code, and a formula was then agreed. (In addition to the substitutable definition and this formula, there are two further defining styles, again both with clear, well-defined syntax and semantics.)

Since concepts are the keys to using the Activator, their names are chosen with great care. All use very common English words that learners are very likely to know (as verified through examination of the Longman Learners Corpus). This makes a marked contrast with Roget which has SATIETY, CONSANGUINITY and TERGIVERSATION (presenting a challenge even to native speakers) where the Activator has ENOUGH, FAMILY and CHANGE YOUR MIND.

There is a small metalanguage used in both concept names and lexicalisations. STH and SB are placeholders for any expression denoting a thing or a person; '/' presents alternatives, the 'house/flat/place etc' construction indicates that the slot may be filled by *house*, *flat*, *place* or words or expressions of similar meaning.

A concept may be lexicalised using a verb, noun or other part of speech: concepts are not specific to one word class. At the level of the section, lexicalisations generally all have the same major grammatical category. At the lexicalisation, the grammar for the word or phrase is presented as in a dictionary such as LDOCE but with the addition of categories for the grammar of phrases and other multi-word units. Here again, the Activator breaks new ground, presenting an account simple and broad enough to allow for the consistent classification of a very wide range of constructions.

There are many other aspects of lexicographic style where the Activator uses proven, familiar LDOCE strategies. To cite just two examples: both the section-level definitions and the definitions of the lexicalisations use only the 2000-word Longman defining vocabulary; and the example sentences are chosen, using corpus data, to show the typical lexical, syntactic and social contexts in which the word is used.

3 The conceptual framework

Much work on taxonomising the language has started from philosophical questions and concerned itself with the abstract upper reaches of the taxonomy. The Activator, by contrast, is a bottom-up enterprise. The work has been at the level of determining which words and phrases fall under the same concept.

Probably the closest relation to the Activator 'concept' is the cognitive psychologists' 'basic level category'. The hypothesis is that there is a level of conceptual organisation that is basic, comparable to the genus in biology (?). There are a variety of indicators as to what is a basic level category, coming from linguistics and cognitive psychology. From psychology, Rosch and colleagues used the number of common attributes subjects can list, associated motor movements, and shape similarity as diagnostics (?). From linguistics, words for basic level categories tend to be short single morphemes, and are generally the first learnt by children (?, ?). Given the Activator's pedagogical purpose, the relationship to approaches to classification based on first-language learning is unsurprising.

The Activator framework bears a limited resemblance to semantic fields. The archetypes

for semantic field theory are 'chair' nouns and cooking verbs. The only entry for *chair* in the Activator is at the concept IN CHARGE OF. The COOK concept has ten sections: the first includes *put sth on, prepare* and *rustle up*, and then the second and third cover the transitive and ergative versions of the semantic field, as traditionally conceived – *cook, boil, simmer, fry, bake, roast* etc.

One point of contact with semantic fields relates to defaults. No definition is given for *steal*, where it occurs as the first item under section 1 of the concept, where the whole section is defined as "to take something that does not belong to you". *Steal* is, naturally, the default lexicalisation for the primary meaning of the concept STEAL. To give another definition for the lexicalisation would be duplication.

Sometimes, corpus analysis and pedagogical orientation will suggest a classification at odds with the semantic field: whereas *rape*, *murder* and *shoplift* may all be in the semantic field CRIME, the Activator puts *rape* at HAVE SEX, *murder* at KILL, and *shoplift* at STEAL.

4 An inheritance hierarchy for NLP and AI

That there is a 'knowledge acquisition bottleneck' has been a truism of artificial intelligence for at least a decade. As the language-understanding components of wide-coverage NLP systems become more ambitious, so this translates into the acquisition bottleneck for lexical semantic information. In AI and NLP, all are agreed that organising words, or concepts, into hierarchies is fundamental to any viable acquisition strategy. Once there is an inheritance hierarchy, many items of information need only be added to a lexicon or knowledge base once. They will then be available by inference for a large number of words or concepts.

A crucial item to acquire, then, is the hierarchical structure. This was the objective of the first major exercise in machine-readable dictionary research (?), and in many that have followed since (see, e.g., papers and references in (?)). For much statistical work, this has again been the goal (?, ?). Machine-readable resources offering taxonomies directly, such as Roget, the Longman Lexicon (?) and WordNet (?) have been widely used (?, ?), but these resources have not been corpus-based.

The Activator is unique in presenting a thoroughly-researched, corpus-based, wide-coverage organisational scheme for the core vocabulary of English. The fact that it does not go beyond low-level organisation is appropriate given the empirical, theory-(relatively)-neutral nature of the resource. For any large exercise in building a knowledge base, the designers will approach the task with their own ideas about the upper part of the hierarchy, and how it must be organised to satisfy their theoretical predilictions regarding semantic primitives and inference mechanisms. The Activator leaves such territory untouched. It leaves 1000 concepts unclassified, at its most general level. This is not an unduly large number of nodes for a knowledge-base builder to hand-craft into his or her lexicon; the objective of using machine-readable dictionaries is to avoid hand-crafting tens of thousands of entries, not to avoid hand-crafting one thousand.

5 A resource for generation

The Activator is designed to help foreign learners of English choose the appropriate word, and use it correctly. While a natural language generation system is not a foreign learner (as commonly understood) it faces the same problem.

Two recent papers, (?) and (?), explore this territory. DiMarco and Hirst discuss the difficulty of choosing between near-synonyms, and review the help on offer from existing dictionaries. They consider the usage notes and synonym essays of existing dictionaries, and determine that these are occasionally useful but have not been prepared in a systematic way, and are in any case not to be found for most words. They conclude with an appeal for a major exercise in lexicography targeted at identifying and distinguishing between near-synonyms. Their paper was written before the publication of the Activator.

Stede also investigates lexical choice, looking specifically at lexical style, defined as "broadly ...the various ways of expressing the same message" (p 455). He breaks down 'style' into a number of dimensions, including FORMALITY, EUPHEMISM, FLORIDITY and FORCE, for each of which a word has a score. A generation system which had, as input, not only the meaning to be expressed but also the formality, floridity, etc. it was to be expressed with, could then find the best match. Stede presents a few examples arrived at subjectively, and says that psychological experiments are needed to find and validate the scales and the scores words have on them. We would claim that, firstly, language corpora offer a more appropriate methodology than psychological experiments, and secondly, that much of the work has already been done (albeit with output written in English rather than numbers) in the course of the lexicography leading to the Activator.

Lexical choice is a central question for natural language generation. Choosing the right word requires knowing the meaning distinctions between near-synonyms. Longman lexicographers have spent five years doing just that, in a carefully structured, corpus-based way. Future work on lexical choice cannot afford to ignore this new resource.

Reference

Amsler, R. A. (1980). *The Structure of the Merriam-Webster Pocket Dictionary.* Ph.D. thesis, University of Texas at Austin.

Beckwith, R., Fellbaum, C., Gross, D., & Miller, G. (1991). WordNet: A lexical database organized on psycholinguistic principles. In Zernik, U. (Ed.), *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, pp. 211–232. Lawrence Erlbaum, Hillsdale, New Jersey.

Berlin, B. (1978). Ethnobiological classification. In Rosch, E., & Lloyd, B. B. (Eds.), *Cognition and Categorization*, pp. 9–26. Lawrence Erlbaum, New Jersey.

Boguraev, B. K., & Briscoe, E. J. (Eds.). (1989). *Computational Lexicography for Natural Language Processing.* Longman, Harlow.

Church, K., Gale, W., Hanks, P., Hindle, D., & Moon, R. (1994). Substitutability. In Atkins, B. T. S., & Zampolli, A. (Eds.), *Computational Approaches to the Lexicon.* OUP, Oxford.

DiMarco, C., & Hirst, G. (1993). Usage notes as the basis for a representation of near-synonymy for lexical choice. In *Making sense of words: Proc. Ninth Ann. Conf. of the UW Centre for the New OED*, pp. 33–43 Waterloo, Canada.

Hindle, D. (1990). Noun classification from predicate-argument structures. In *ACL Proceedings, 28th Annual Meeting*, pp. 268–275 Pittsburgh.

Lakoff, G. (1987). *Women, Fire and Dangerous Things.* University of Chicago Press.

McArthur, T. (1981). *Longman Lexicon of Contemporary English.* Longman, Harlow.

Proctor, P. (Ed.). (1978). *Longman Dictionary of Contemporary English.* Longman, Harlow.

Rosch, E., & Lloyd, B. B. (Eds.). (1978). *Cognition and Categorization.* Lawrence Erlbaum, New Jersey.

Sanfilippo, A., & Poznanski, V. (1992). The acquisition of lexical knowledge from combined machine-readable dictionary sources. In *Proc. Third Conf. on Applied Natural Language Processing*, pp. 80–87 Trento, Italy. Association of Computational Linguistics.

Stede, M. (1993). Lexical choice criteria in language generation. In *ACL Proceedings, 6th European Conference* Utrecht, Holland.

Summers, D. (Ed.). (1987). *Longman Dictionary of Contemporary English, New Edition.* Longman, Harlow.

Summers, D. (Ed.). (1993). *Longman Language Activator.* Longman, Harlow.

Yarowsky, D. (1992). Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *COLING 92* Nantes.

Facilitating the Corpus-Building Process and Maximising the ,,Analytical Yield'': A LSP-Oriented Case Study

FRANK KNOWLES - PETER ROE

ABSTRACT

This paper reports on accumulated experience vis-a-vis LSP lexicography and its associated computational techniques gained during research carried out on large LSP (i.e. Language for Specific Purposes) text corpora in English, French and German. In this particular case study text selections were made (with the copyright holders' full permission) from the CD-ROM version of the "Financial Times" (FT) — parallel to studies of "Le Monde" and "Handelsblatt". The approach used in the research described indicates the possibility of shifting the ground of lexicography away from its traditional notional centre of gravity to an AI-oriented strategy for capturing genuine cognitive units without robbing them of their textuality. This leads naturally to a powerful, qua direct, type of lexicographical codification, the primum mobile of which is not definition, but distribution.

Introduction and Definitions

The research reported on in this paper is the direct result of investigations into the field of LSP (Language for Specific Purposes) undertaken by the LSP Research Sector within the framework of the Institute for the Study of Language in Society at the University of Aston. The LSP concept, increasingly important in the context of the current rapid process of internationalisation, is defined in terms of: language, discourse and community, as follows:

- The term "language" is here reserved for what is covered by the set of published conventional dictionaries and grammars of a national or international language, e.g. Hungarian or French, while "discourse" is used to refer to the subset of the options offered by a given language which are actually adopted systematically by a given community;

- The term "community" is here used to refer to any group of individuals who are defined by a shared global purpose to which all publicly subscribe, and who have evolved or adopted mechanisms and procedures for achieving their shared objectives. Examples are: a Board of Governors, subscribers to a journal on power transmission, or suppliers of computer equipment and their end-users;

- The term "discourse community" is used to refer to a community as defined above with particular regard to the discourse used by its members in the pursuance of its goals. For further discussion of this term, see Swales (Swales J, Genre analysis: English in academic and research settings, CUP, [1990]. See also in this connection: Nystrand M, The structure of written communication: studies in reciprocity between writers and readers, Academic Press, [1986] and Johns A M, "L1 Composition theories: implications for developing theories of L2 composition" — in: Kroll B, Second language writing: research insights for the classroom, CUP, [1990]).

- At Aston the LSP Research Sector further, and more narrowly, defines LSP as the effective response to two related questions posed by a specialist in a given field wishing to operate in that area of specialisation within a community using an as yet insufficiently familiar language. (For further discussion see: Roe P J, *User Modelling in CALL: Some Fundamental Issues*, in Kibby M (Ed.), Computers & Education: An International Journal, Pergamon, [1994, forthcoming]). The *primum mobile* questions adumbrated above are as follows:

  Firstly, what do I need to "know", that I do not already "know", in order to be able to operate efficiently within the new Discourse Community?

  and:

  Secondly, how may I most efficiently make up this (linguistic) deficiency?

A significant part of the problem facing any professional person seeking valid answers to these questions concerns what constitutes "knowledge" in the above sense. The crude misconceptions often adopted have led to gross inefficiencies in the past in meeting learner needs and providing solutions that are efficacious, effective, efficient and affordable. One of the worst incongruities is ignorance on the part of the "teachers" of the target discourse. It very rarely happens that they are themselves members of the target Discourse Community. Few make the effort to gain entry to observe for themselves, and of those who do make the attempt, many are turned away by the gatekeepers on grounds of confidentiality. The common solution is to have recourse to "language", as defined above, with perhaps a flavouring of lexis commonly associated by outsiders with the pre-occupations of the Discourse Community concerned. This gives rise to such notions as "Business English", or "Scientific English", terms so vague and general as to be practically meaningless. These are language, not discourse, terms.

Other serious inefficiencies arise from our current inability to provide rigorous answers to the above two "learner questions" which are not the prime concern of this paper or this conference. We shall here pursue the issue of defining what the learner needs to "know" in terms of the discourse of the target community, and examine the implications for the lexicographer of the 21st Century.

Dictionaries and Discourse Communities

The heart of the problem can readily be appreciated by reference to the prevalent use of what Chambers (The Chambers Dictionary, Chambers Harrap [1993], page xi) calls "Classifications labels". These are labels "relating to the classification (e.g. *colloq, slang, chem, elec, psychol*) of a word or meaning". These classifications are obviously of two types, one represented by "*colloq, slang*", and the other by "*chem, elec, psychol*". There are 72 exponents of the latter set in the list of abbreviations on pages xiv to xviii, and many more unabbreviated examples in the main text (we counted over thirty). Now these labels relate to fields of human knowledge or enquiry, and these must surely number vastly more than the hundred odd listed in this dictionary. But the true number must be insignificant when compared with the number of sub-fields which these subsume, such that an expert in one is unlikely to be also an expert in any other. But this complexity is further compounded by the fact that the discourse of communities (as defined above) operating within these fields and sub-fields will be determined primarily by purpose (their respective "agendas"), and the mechanisms and procedures and relationships which they have developed, rather than by field. And this discourse will be further modified by community rules governing categories in the other set, which one might loosely designate 'style and tenor', e.g. "colloq, slang, euphem, vulg, formal, derog, taboo, facetious" etc. etc.. So although all communities will dip into the same pool (language) for their linguistic resources, no two will do it in identical ways. It is a common enough experience for native speakers to suffer adaptation trauma when moving from one Discourse Community to another, e.g. someone taking the witness stand for the first time, or a 'family' bridge player agreeing to make a fourth in a serious competition match. (Witness the evidence of a non-linguist bridge-player, Tony Forrester, writing in the "Daily Telegraph" on Saturday April 16 1994. "Each field of endeavour has its own unique terminology. ... Bridge carries an arsenal of such terms, generally to define some unusual play or defensive counter. ...".)

The biggest indicator of the reasons for the inadequacy of dictionaries generally to meet the knowledge requirements implied by the learner questions given above is to be found in the give-away phrase on page xi: "..... where there is only one sense and one meaning of a word". This seems to suggest that there exists a well-defined set of meanings and senses, onto which is mapped an, albeit smaller, set of words. This position we would designate classical, or Newtonian, semantics, by analogy with classical, or Newtonian, mechanics, with which one can get by perfectly adequately until the day when more searching and demanding questions are asked. Our two learner questions can be answered only by reference to an adequate theory of 'quantum' semantics, if one could but find the right analogy for the solution arrived at by Planck in the case of mechanics.

The view advanced here takes as its starting point, not the lexicon and a dissection of its elements into meanings and senses, a procedure long discredited by, for instance, Bolinger, but the plethora of Discourse Communities which make up the human race, many unstable, ephemeral, all driven by wants, need and ambitions, whose 'business' is mediated through mechanisms adopted from other contexts and adapted to suit the character of the new environment. But the values and notions, the "meanings" and the "(sub-)senses" which constitute the world of the community, are in constant flux. The closed set of lexical items available is (relatively) impassive, indeed inert, and is ill-suited to reflecting changes in sense by changes to its members. This closed set is as powerful and flexible as the systems of Newtonian mechanics; the world of meanings is as complex and as infinitely variable as the quanta of Planck. The members of the community, in order to be able to treat the new meanings their world has generated, are faced with a choice of selecting a 'best fit' from the existing lexicon, introducing a neologism, inventing a nonce-word, or visiting a new meaning on an

extant item without the slightest suggestion of semantic overlap (e.g. quark, charm, strange etc. in the case of subatomic particles). In either case the new sub-sense must be negotiated with the other members of the community. (For a fuller discussion of meaning negotiation see Knowles F and Roe PJ: *The Way Words Work Together — LSP and the notion of distribution as a basis for lexicography*, Sixth EURALEX International Congress, [1994, forthcoming]). We offer the well-known example of the Discourse Community of what used to be called in English "dustmen" together with their employers and customers. The "dustmen" have negotiated with other members of their communities a variety of terms chosen to reflect new perceptions and values. Language (for Hesse "unsere arme Idiotensprache") responds but crudely, and an adaptation ("purification") is made to Eliot's "dialect of the tribe". Consider likewise the Discourse Community of British Rail and what used to be called its passengers. BR is now negotiating (by force majeure!) the replacement of the term "passengers" by "customers". These are but obvious public examples of the kind of negotiations constantly taking place to some degree or other throughout the entire plethora of Discourse Communities. Meanings are many, words are few. Meanings are visited on words, the shells, the husks of meaning, like the ugly sister's foot into Cinderella's shoe. Under ordinary ("classical") circumstances the degree of misfit is immaterial, and nobody notices. But for our LSP learner, hungry for parole but being fed on langue, the difference is crucial.

The learner questions under discussion can thus be seen to subsume the following questions revolving around down-stream 'quantum semantics'

> Firstly, how does my target Discourse Community map the concepts with which I am familiar into — not onto! — language items?
>
> Secondly, what are the discourse characteristics of the genres associated with my discipline?

The move towards ever-more-specialised lexicons is one 'classical' step in the right direction. And the increasingly popular practice of citations borrowed from the dead languages, especially under the influence of A S Hornby, and accelerated by the COBUILD movement, is another. But they can never effectively meet the requirements of the work-embedded LSP learner, any more than classical mechanics could answer the questions facing Planck and his contemporaries. The dictionary gloss can never remotely hope to provide the kind of 'knowledge' our learner requires. However counter-intuitive at first sight, the answer can lie only in 'dictionaries' constructed out of exponents of the genres the learner will have to manipulate in reality.

The rest of this paper explores an approach to the solution of this problem as far as written text is concerned, and discusses the general implications arising out of this.

The Aston Scientific and Technical Corpus (ASTEC)

This corpus was originally conceived along classical lines, as consisting of careful and measured selections from representative samples of technical literature. It soon became apparent, however, that the researchers who wished to make use of the corpus were always interested in a Discourse Community for which no text was available in the corpus. So a new sub-corpus had to be created each time. Once this had been accepted as the norm, the emphasis shifted to the provision of facilities for the rapid creation and analysis of ad hoc corpora, which is what ASTEC now is. Two platforms are used. One is based on a variety of dialects of UNIX on SUN workstations which will rapidly process the largest files we expect to have occasion to use for LSP purposes. The order of magnitude of corpus size found useful is 100,000 for a well-defined, narrow-focus Discourse Community, rising to 5,000,000 in the case of a more diffuse one, such as the readership of the Financial Times, which is the example taken in this instance. These facilities enable the LSP learning programme facilitator to establish what is most characteristic of the corpus, in the light of the two learner questions, with reasonable ease and rapidity. However, UNIX, although extremely fast and flexible, is not the most user-friendly or easily transported system, and consequently Aston is arranging for a C++ version,

ATA [the **Aston Text Analyser**], to be mounted under MS/DOS, in collaboration with our technical/commercial partner MS Technology A/S Copenhagen. The ATA prototype is currently called from a PIF under Windows 3.1.1, but will eventually be developed to operate under Windows' successor.

This move from UNIX to C++ occasioned a complete rethink of the approach to the problem of producing a pedagogic description of a target corpus, suitable for output in a form appropriate to the LSP learner's second question. Instead of producing and storing and further searching large files of the analysed text, it was decided instead to create a database via which the entire corpus could be interrogated directly to produce whatever data the facilitator thought fit. The time taken for the creation of the database was not critical, but the accessing and outputting facilities could not be allowed to involve serious wait time.

We will now illustrate the facilities of ATA and ASTEC, and also use the sample results to illustrate the points made in the introductory section. The overall corpus for this purpose is constituted by the CD-ROM edition of the Financial Times 1992 (kindly supplied, for research purposes, by the Times Newspapers). A headline search yields the initial breakdown shown in Figure 1.

Analysis by Types and Tokens

Focus	Articles	Tokens	Types	To/Ty	Hapax
Full Year	64,414	?25,000,000			
14 Days	3,142	1,150,869	42,915	27	18,111
Stock Markets	2,082				
London		762	526,537	13,397	39	4,306
World		1,320	663,812	16,569	40	6,549
Money	1,895				
M Markets		359	156,883	7,309	22	3,226
Foreign Exch		291	124,618	5,665	22	2,298
Govt Bonds		222	137,058	5,318	26	2,035
Int. Capital		1,023	320,330	15,169	21	6,227
Companies	12,834				
UK		6,613	1,566,726	32,630	48	12,050
International		6,221	1,712,617	35,204	49	12,930
Commodities	1,770				
World		444	62,007	4,067	15	1,665
Agriculture		1,326	551,926	211,552	26	8,511
World News	2,309		130,065	14,560	9	6,863
All Topics	20,890		5,952,579	67,004	89	24,381

Figure 1. An analysis of the tokens and types found in the Financial Times. [Hapax is short for 'hapax legomena', types with a frequency of 1.]

Altogether 5,949,325 tokens and 67,004 types were analysed. Illustrations below are taken from one of these sections, namely the "World Stock Markets" Corpus (663,812 tokens). Times quoted are for ATA mounted on an IBM PS/2 Model 90 XP 486.

Frequency lists

Once the database has been generated, frequency lists of 16,569 lines, in frequency or alphabetical order, can be summoned in a matter of about 2 to 5 seconds each, and then held in memory for instant access. The lists show raw frequency, relative frequency (out of 10,000), the relative frequency of the same type in a lists derived from COBUILD (see Roe P J, "Astec: User's Guide to the Aston Corpus of Scientific and Technical English", Language Studies Unit / Aston University, revised 1994), and an evaluation of the significance of the difference of these two. Further sub-lists can be generated by means of a string search. An example is shown in Figure 2.

6	amounting	0.08	0.02	1.38
9	amounts	0.13	0.27	0.00
1	ample	0.01	0.12	0.00
1	amplified	0.01	0.03	0.00
41	amr	0.61	0.00	8.32
1	amr's	0.01	0.00	7.31
10	amro	0.14	0.00	9.61
1	amsouth	0.01	0.00	7.31
251	amsterdam	3.76	0.03	4.85
17	amsterdam's	0.25	0.00	10.14
1811	an	27.13	33.09	1.60
4	anaemic	0.05	0.02	1.09
2	analysed	0.02	0.06	0.00
2	analysers	0.02	0.00	8.00
3	analyses	0.04	0.05	0.00
25	analysis	0.37	0.61	0.00
192	analyst	2.87	0.03	4.58
4	analyst's	0.05	0.00	8.69
804	analysts	12.04	0.03	6.02
4	analytical	0.05	0.04	0.00
2	anamint	0.02	0.00	8.00

Figure 2. Part of an ATA frequency list of Corpus B, in alphabetical order. The columns contain: Raw frequencies, the type, the relative frequency of the type (out of 10,000) the relative frequency of the type in an adapted COBUILD list, and a significance figure based on columns 3 and 4.

The attention of the facilitator would normally be attracted by the significance figures for the lemmas *analys+*. Now it has been argued elsewhere (Knowles F and Roe P J [forthcoming, as above] and Geffroy A, Lafond P & Tournier M, ERA 56 au CNRS, ENS de Saint-Cloud, [1973]) that lemmatisation may hinder rather than help the LSP learner, as it is an 'etic' rather than an 'emic' phenomenon. This argues that the *analys+* lemmas do not necessarily share — and most probably do not share — a unique semantic value, say ANALYS. ERA 56 (Geffroy A, Lafond P & Tournier M, [as above]) even argue (convincingly):

"La confusion du singulier et du pluriel sous la même forme canonique est néfaste du point de vue statistique." op cit p. 21

Concordances

We likewise see lemmatisation as potentially seriously misleading. The various types forming an individual lemma constitute independent meaning shells, each of which can be charged with meaning independently of the others. Polysemy is nothing new, but it is still commonly assumed that the addition of the plural *+s* and the adverbial *+ly* leave the underlying value intact in some significant way. One obvious case attested by the present corpus is that of *sure* and *surely* not sharing a sememe SURE (see Knowles F and Roe P J [forthcoming]). In the above case of *analys+* (Figure 2), further inspection by means of concordances suggests that caution is necessary if we are not to mislead the LSP learner. Operationally, ATA produces concordances of over 2,000 lines per type in 15 to 20 seconds, providing left or right ordering virtually instantaneously. Extracts are shown in Figure 3.

Synoptic Profiles

However, for the LSP facilitator much the most important ATA tool is the synoptic profile generator. This provides frequency listings for all types occurring in the range -3 to +3 of the type in question. Figure 4 shows the frequency listings for *blue* and *chips*, and *blue-chip(s)* suggests the possibility of a close relationship. Scrolling through concordances to establish the

connection would take some time, whereas the synoptic profiles demonstrate that *blue* and *chip* have virtually no separate and independent existence of their own, as is evident from Figure 5.

```
          share in auto emission  analysers in
    maker of automotive emission  analysers, surged Y260 to
                                - -
                         SBC has  analysed 60 companies which make
                    Merrill Lynch analysed the company at the
                                - -
               Mr Guido Meier, an analyst at UBS in Zurich.
                     119 3/4 as an analyst raised
                              An  analyst said that Allianz might
                  to end flat. An analyst said the market
           after heavy trading. An analyst said there is a
```

Figure 3. Extracts from some of the concordances for *analys+*, showing clearly the semantic contrast between the use of *analysers* and *analyst*.

Likewise, it is easy to show the community-specific values, and indeed morphosyntactic constraints, peculiar to this kind of corpus. Thus the frequent expression *over-the-counter* has no independent existence, and would never constitute part of the answer to the LSP learner's questions unless linked specifically to the few items shown in Figure 6. The corpus is full of such restricted items. See also Figure 7 for *sentiment*, a word which can be considered as having no existence other than in the sense implied by *market sentiment*. The LSP learner would be only hindered by any suggestion that the word has other meanings, or that *market sentiment* is in fact a derived meaning.

408	blue	6.11	1.30	2.19
25	blue-chip	0.37	0.00	10.24
12	blue-chips	0.17	0.00	9.79
-	-	-	-	-
131	chip	1.96	0.07	3.36
1	chipcom	0.01	0.00	7.31
283	chips	4.23	0.10	3.82

Figure 4 Frequency listings for "blue" and "chip".

blue					
11 was	19 High	35 of	266 chips	32 were	37 the
11 to	17 the	26 in	130 chip	31 stocks	11 Straits
11 of	15 buying	23 technology	6 fin	25 The	10 as
9 the	12 number	17 the	2 skies	21 issues	10 The
8 a	11 in	14 internati		18 and	9 were
7 in	11 Among	14 for		10 which	8 was
chip					
10 number	17 of	130 blue	33 stocks	10 The	12 the
8 the	13 the	16 Blue	23 issues	7 which	4 to
4 of	9 in	2 computer	9 electric	7 were	3 up
4 in	4 by		7 indicator	6 index	3 rose
4 a	4 buying		7 index	4 the	3 ground
chips					
19 High	23 technology	266 blue	36 were	39 the	16 the
12 buying	19 of	23 Blue	25 The	11 Straits	11 on
8 the	17 in	2 BLUE	19 and	10 as	11 Times
8 in	14 internati		12 which	7 in	8 index
8 demand	12 Among		9 in	6 to	5 with
8 Among	11 industrial		8 with	6 on	4 their

Figure 5. Synoptic profiles for *blue* + *chip* and *chips*. The columns contain raw frequencies and type.

52 Nasdaq	51 composite	70 of	74 stocks	5 slipped	9 2
17 composite	32 On	51 the	53 market	5 however	8 the
6 listed	18 index	6 In	11 trading	5 fell	7 3
5 4	13 on	4 and	2 Malaysian	5 Microsoft	6 1
4 2	5 in	4 active		5 Intel	4 jumped
3 second	3 section	2 in		4 which	4 5
2 traded	2 2	2 Among		4 was	3 declined
2 the				4 gained	3 at
2 share				4 climbed	3 8
2 its				3 dropped	3 4

Figure 6. Synoptic profile for *over-the-counter* (f=147). The columns contain raw frequencies and type.

16 the	39 to	42 market	47 was	51 the	23 by
15 to	22 also	28 on	44 The	12 positive	22 index
15 failed	16 on	28 depressed	33 remained	8 said	11 was
9 continued	13 of	17 in	29 in	8 also	11 the
8 survey	11 said	16 that	13 and	7 nervous	8 market
6 on	9 the	15 lift	8 had	7 a	8 in
6 of	8 that	14 as	8 as	6 weighted	8 a
6 little	8 change	11 business	6 which	5 depressed	7 The
6 in	7 weighed	10 investor	6 on	5 by	6 to
6 helped	7 stock	9 positive	6 has	5 been	6 that
5 a	7 in	9 lifted	5 remains	4 still	6 ahead
4 said	7 for	8 but	5 is	4 of	5 were
4 influence	7 effect	8 and	5 However	4 firm	5 of
4 but	5 trading	8 affected	5 By	4 dampened	4 stock
4 benchmark	5 improvement	8 Market	4 writes	4 composite	4 on
4 an	5 and	6 underlying	4 with	4 Tokyo	4 investor
4 also	4 influence	6 negative	4 -	3 were	4 from
3 was	4 helped	6 helped	3 towards	3 trading	4 as
3 shift	4 dollar	6 dampened	3 to	3 to	4 and
3 interest	4 depressed	6 consumer	3 than	3 some	3 positive

Figure 7. Synoptic profile for *sentiment* (f=404). The columns contain raw frequencies and type.

"Chunking"

But such synoptic overviews are of even greater value for the LSP course designer, or indeed learner, when it comes to encapsulating what the learner needs to "know" about syntactic and collocational patternings, or "chunkings" (Skehan P, *Second Language Acquisition Strategies and Task-Based Learning*, in: Thames Valley Working Papers in English Language Teaching, Vol. I, [Spring 1992]). It is clear from Figure 6, for example, that *over-the-counter* is bonded to *stocks/market/trading*, and that these are in turn bonded to a small set of expressions of change. From the LSP learner's point of view, it is a sufficiently safe assumption to say that for this particular corpus such a description is a powerful generalisation. Similarly *sentiment* in Figure 7 is always *market sentiment*, with the word *market* being commonly deleted or substituted by one of a small set of related items. These in turn are followed by one of a small set of indicators of change or status quo. Similarly, *composite*, in Figure 8, can be seen to form part of the "chunk" [NAME OF STOCK EXCHANGE] + *composite* + [INDEX (implied or stated)] + [VERB OF MOVEMENT].

John Sinclair has stated recently (during a lecture at the International Symposium on Phraseology in Leeds in April 1994) that whereas the central columns in concordances represent langue, the horizontal lines constitute parole. And it is parole that the LSP learners are seeking. But these parole data need to be made accessible to them in some way. The synoptic profile is seen as a most useful tool in the hands of the learning programme facilitator.

129 while	326 the	409 The	729 index	60 fell	52 stocks
128 and	45 The	254 Nasdaq	52 of	58 rose	49 2
79 the	36 and	185 the	22 was	57 closed	48 4
25 on	36 American	75 Amex	16 rose	55 gained	48 3
20 prelimin	35 trading	36 SE	15 put	54 1	46 1
20 The	33 as	29 300	14 stock	51 over-th	41 0
14 but	25 profit-t	21 LUMPUR's	14 0	45 ended	38 5
13 of	24 TSE	20 KLSE	13 lost	42 lost	37 up
11 moderate	21 KUALA	14 SEOUL's	13 fell	41 was	30 6
10 with	16 data	13 MANILA's	13 eased	38 shed	30 10
10 trading	12 with	7 TSE-300	13 added	38 0	23 8
10 market	11 shares	3 TSE	12 slipped	31 2	23 12
9 stocks	10 market	3 300-stock	12 ended	25 up	20 on
9 light	8 week	2 US	11 up	21 added	20 9
8 profit-ta	8 trade		9 closed	20 dipped	20 7
8 president	8 low		8 gained	20 climbed	18 down
8 bargain	8 hunting		7 shed	19 put	18 16
7 to	8 York		7 gave	19 of	18 13
7 recent	7 stocks		7 firmed	17 slipped	17 over-th

Figure 8. Synoptic profile for *composite* (f=1,083). The columns contain raw frequencies and type.

We now turn to a general discussion of just what help the lexicographer can be in this task.

Discussion and Conclusions

"We must never try to define the meaning of a word in isolation, but only as it is used in the context of a proposition."
Translation from: G. Frege, Die Grundlagen der Arithmetik, Koebner, Breslau, 1894

"Only propositions have sense; only in the nexus of a proposition does a name have a reference."
L. Wittgenstein, Tractatus Logico-philosophicus, Routledge & Kegan Paul, London, 1923

The LSP investigations reported above (confined, in our case, to written text at this stage) can be classed on a focal cline ranging from the extremely narrow (e.g. the language of pilots in the air or at sea) to the extremely broad (e.g. the English novel). On this cline the language of the various domains of the FT ranges from narrow to mid-focus. For those who pursue this line of research, meaning and value are seen as properties of a discourse community negotiating its proper outcomes. Thus these meanings/values seek vehicles for their expression and find "best fits" in the shared linguistic experience of other discourse communities. This view contrasts sharply with meaning seen as a pre-determined, inherent and even inherently stable property of lexis. One conventional approach is the "fractionating" of lexical items into their world of possible "meanings". The less conventional stance adopted here is to conceive of "meanings" as peculiar to the context in which they arise. Such "meanings" then innovatively attach themselves to convenient extant lexical items or they realise themselves in neologisms. Thus, whereas the world of the lexicon is relatively stable, the world of meanings is inherently unstable. The mapping of new meanings onto old lexis is inexact and has a considerable aleatory dimension.

The approach taken by this research is not to tread the path of notional definition and categorisation but rather to determine what (relatively arbitrary) decisions a particular discourse community makes and how it "valorises" the concrete choices which flow from such decisions. All we can be sure of are the lexical selections actually made and their distribution. The rest —

that is, the traditional work of lexicographers — must contain an element of guesswork, admittedly often seemingly inspired guesswork as the attempt is made to establish equivalences which at best can only be partial and at worst highly misleading and even dangerous. However, it is only the advent of available and affordable technology, coupled with copious sources of rich data, that enables us to make statistically significant statements of a more objective nature. In this case, it is the FT's and analogous stratified and encyclopaedically bound and demarcated material, manifest in several "sub-genres", which can be safely classified as typical of the use of language for specific purposes (LSP).

Software routines especially constructed for this type of macro-corpus analysis, but not relying on any type of lemmatisation or tagging, are able to identify and extract — with considerable rapidity, flexibility and transparency of access because of an ability to hold millions of running text words in a dynamic index — the essential phraseology and terminology of this global discourse domain, potentially cascading it into appropriate bins representing authentic sub-domains. The phenomenon of semantic and/or encyclopaedic scatter or vagueness normally likely to occur during such a process is either eliminated altogether or is reduced — should it occur — to entirely manageable proportions by the chunk approach as opposed to the routine extraction of orthographic words. The actual meaningful "chunks" identified in this way are clustered and hence made amenable to lexicographical-cum-terminographical treatment. These chunks — either single or multiple orthographic words, flush or discontinuous — may label fully-fledged "professional" cognitive units, everyday cognitive units, fixed, stable or variable collocations of either a universal or domain-specific nature. Frequency tabulation provides a strong initial basis for classifying them one way or another.

A natural consequence of the ability to identify the cognitive units comprising a text is the emergence of the possibility and desirability of quasi-lexicographic definition, based on an AI-oriented strategy, which does not rely on any notional basis but proceeds operationally — just as text itself does — to invoke or excite appropriate cognitive structures in readers' minds. This is perhaps best understood as a non-interventionist, non-interrogative, non-interactional analogue of Expert System techniques for eliciting professional expertise. All told, we have, via the above technique, a method of achieving a type of definition of professional cognitive units not by traditional methods, but rather by distribution, notably by the co-locational and collocational dynamics of the linguistic "names" which refer to them. Tracking the frequency and sequencing of the instantiations of these professional cognitive units is, of course, an option not without interest or significance in its own right. The dynamic definitions produced by the above routines are not robbed of their textuality but are progressively modulated by their environment. Normally, lexicographers provide only "final", not developing definitions. Above all, the approach discussed here deals with genuine cognitive units, not with abstractions or fusions of them. In this way a vital bridge is created between static, extra-textual dictionary (sub)-senses and dynamic textual meaning(s). It also follows from the above that considerable scope also exists for the automatic disambiguation of homographs and of sub-senses within a lexeme.

What is perhaps of particular interest to COMPLEX94 is the claim that the approach used in the research described indicates the possibility of shifting the ground of lexicography away from its traditional notional centre of gravity to an AI-oriented strategy for capturing genuine cognitive units without robbing them of their textuality. This leads naturally to a powerful, qua direct, type of lexicographical codification, the primum mobile of which is not definition, but distribution.

In this general way what might be called the "analytical yield" of statistically- and (juxta)positionally-oriented "probes" into the corpus material is maximised. Within the gross subject domain (set here a priori as finance) thematic markers and terminological usages achieve a largely automatic but highly significant lower-order structuring and stratification of the corpus material on an *ex post facto* basis.

Lexicographie Computationnelle et Auxiliaires des Langues Romanes

BÉATRICE LAMIROY

This paper shows how a perspective of computational lexicography, in which extremely systematic descriptions and exhaustive codification of the data are necessary, may uncover unexpected differences between related languages. The data examined here belong to the class of so-called (semi)-auxiliaries and are taken from French and Spanish. They are described according to Gross's model of lexicon-grammars. Whereas in French only few verbs of this class subcategorize for both an infinitival and an NP complement, in Spanish the latter property holds for more than half of the verbs. It is claimed that this empirical difference between the two verb classes is not only relevant from a typological point of view, but also raises the theoretical question of how the universal category of auxiliaries ought to be defined.

O. Introduction.

Le but de cette communication est double. Je voudrais montrer, d'une part, que la codification systématique des données, nécessaire dans une perspective de lexicographie computationnelle, permet de mettre le doigt sur des différences syntaxiques importantes et auparavant insoupçonnées entre deux ou plusieurs langues. J'illustrerai ce point au moyen de la classe dite des (semi-)auxiliaires du français et de l'espagnol, décrite selon le modèle du lexique-grammaire de M.Gross. D'autre part, je m'attacherai à montrer comment un problème posé dans un secteur appliqué de la linguistique, la lexicographie computationnelle en l'occurrence, peut amener à réexaminer une question théorique, à savoir la définition "universelle" de ce qu'est un auxiliaire.

L'exposé est organisé en trois parties. Je commencerai par situer brièvement le problème. J'aborderai ensuite les données typologiques qui opposent le français à l'espagnol et je traiterai pour finir la question théorique concernant la définition de l'auxiliarité.

1. Le problème.

Grâce à de très nombreuses recherches en syntaxe consacrées à la complémentation verbale, les linguistes ont pu se faire une idée assez précise quant à la nature des compléments pouvant être sélectionnés par un verbe. Les possibilités peuvent être ramenées, en gros, à trois types fondamentaux.

Un très grand nombre de verbes sélectionnent un syntagme nominal (ou plusieurs syntagmes nominaux), à l'exclusion d'une complétive ou d'une infinitive, et forment des phrases simples (Boons et al. 1976, Guillet & Leclère 1992). Le syntagme nominal (désormais SN) peut être direct, indirect (en *à* ou *de*) ou prépositionnel :

(1) a. Max boit une bière
b. Max ressemble à son père
c. Max opte pour la fuite

Une deuxième catégorie de verbes forment au contraire des phrases complexes : ce sont les verbes dits à complétive. [1] Leurs effectifs sont plus réduits que dans le cas précédent, mais

---

[1] Les verbes qui sélectionnent une interrogation indirecte (p.ex. *se demander si P*) peuvent être assimilés à cette catégorie. Le fait de prendre une complétive n'exclut évidemment pas pour de nombreux verbes d'avoir aussi un SN dans leur valence, ainsi le type N_0 V que P à N_1 : *Max dit qu'il fera beau à Luc* .

toujours très nombreux (Gross 1975). La complétive peut être directe ou indirecte : [2]

(2) a. Max veut que tu démissionnes
b. Max tient à ce que tu sois présent
c. Max doute de ce que tu sois honnête

Pour les verbes de cette catégorie, la sélection d'une complétive est corollaire de deux hypothèses qui remontent à la grammaire traditionnelle et qui font donc partie de l'acquis en syntaxe. [3] D'une part, la complétive et l'infinitive sont des variantes combinatoires : le choix de l'une ou de l'autre de ces structures est déterminée par l'identité éventuelle des sujets. Et d'autre part, la complétive est de nature nominale, d'où l'étiquette traditionnelle de subordonnée "substantive". Contrairement aux verbes de la première catégorie qui sélectionnent *exclusivement* des SN, les verbes à complétive n'excluent donc pas a priori l'alternance avec un SN :

(3) a. Max veut ta démission
b. Max tient à ta présence
c. Max doute de ton honnêteté

Cependant, on s'est en général beaucoup moins intéressé en syntaxe aux rapports entre les complétives et les compléments nominaux, ces rapports paraissant d'ailleurs irréguliers, même incohérents. Ainsi on pourra opposer aux paradigmes de *vouloir* et de*confirmer*, par exemple, ceux de *espérer* et de *affirmer* respectivement :

(4) a. Max veut (que tu démisionnes + ta démission)
b. Max espère (que tu démissionneras + * ta démission)
(5) a. Max confirme (qu'il part + son départ)
b. Max affirme (qu'il part + * son départ)

Il existe un troisième groupe de verbes qui se distingue des deux premiers, tout d'abord parce que ses effectifs sont beaucoup plus réduits. Syntaxiquement, les verbes de cette catégorie se font suivre d'un complément infinitif, mais la complétive correspondante est exclue. Il s'agit de la classe des verbes traditionnellement appelés (semi)-auxiliaires :

(6) Max (doit + vient de + se met à) parler

L'absence de complétive fait une double prédiction, l'une empirique, l'autre conceptuelle. En

---

[2] En français, les prépositions autres que *à* et *de* n'introduisent que très rarement des complétives. Grevisse-Goosse (1993 : 1601) mentionne *en ce que* (p.ex. *consister*) et *sur ce que* (*insister*).

[3] Elles ne sont donc guère remises en question. Une des rares études (à ma connaissance) à avoir examiné de façon systématique l'alternance complétive-infinitive est Lemhagen (1979).

effet, puisque la complétive est de nature nominale et qu'elle est exclue, on peut s'attendre à ce que le SN soit exclu aussi. Empiriquement, l'absence de SN se vérifie en général en français, mais pas toujours :

(7) a. Max vient de (critiquer ce travail + * une critique de ce travail)
b. Max finit par (critiquer ce travail + une critique de ce travail)

Mais ici encore, les rapports entre infinitives et compléments nominaux sont mal connus: dans la littérature on se contente souvent, comme on le fait d'habitude avec des "irrégularités", d'enregistrer simplement ces cas comme des exceptions. [4]

L'absence de complétive pour ces verbes a une deuxième conséquence, de nature plus théorique. Puisque la complétive et le SN sont absents, l'infinitif serait purement de nature verbale et formerait avec le premier verbe un "complexe verbal" (Rojo 1982). Autrement dit, ces verbes entrent donc - contrairement à ceux de la deuxième catégorie - dans des phrases simples. [5] Par ailleurs, leur capacité de sélection, faible ou nulle, [6] serait révélatrice d'un processus de grammaticalisation (Closs Traugott & Heine 1991) qui les situe entre les verbes lexicaux, à capacité de sélection forte, et les morphèmes grammaticaux, qui en sont totalement dépourvus.

J'aborderai ici les verbes de la troisième catégorie, et en particulier la question des SN pouvant apparaître après eux. Cette question peut paraître accessoire en français, mais elle devient incontournable en espagnol pour une raison statistique : alors qu'en français, les auxilaires qui admettent un SN à la place d'un infinitif correspondent à moins d'un tiers de la table 1 de Gross (1975 : 234), en espagnol la propriété vaut pour plus de la moitié de la classe, p.ex.:

(8) a. Max se detuvo mucho en (explicar los hechos + la explicación de los hechos)
Max s'arrêta beaucoup dans (expliquer les faits + l'explication des faits)
'Max expliqua longuement les faits'
b. Max se agobia por (educar a sus hijos + la educación de sus hijos)
'Max se donne beaucoup de mal pour (éduquer ses enfants + l'éducation de ses enfants)'

---

[4] Une autre solution, depuis Perlmutter (1970), est de considérer qu'il s'agit de deux verbes homonymes.

[5] En termes harrissiens, ils correspondent à des opérateurs U qui prennent une proposition pour argument. Le résultat est une phrase simple.

[6] Une des propriétés des opérateurs U chez Harris est d'être "transparents aux relations syntaxiques et sémantiques sujet-verbe dans lesquelles ils s'insèrent " (Gross 1975 : 161).

2. Les auxiliaires français vs espagnols.

Pour les deux langues, j'aborderai successivement le nombre de verbes qui entrent dans la classe, les prépositions qui introduisent l'infinitif, le type de SN qui alterne avec l'infinitif et enfin, les propriétés du sujet N_0. Tous les verbes dont il sera question, qu'ils soient français ou espagnols, ont la propriété syntaxique de se faire suivre d'un infinitif, alors que la complétive correspondante est exclue : c'est la propriété définitoire des tables 1 du français (Gross 1975) et de l'espagnol (Lamiroy 1991, Subirats 1987). Il existe donc une table équivalente dans les deux langues, mais il existe aussi des divergences importantes qui opposent la table du français à celle de l'espagnol.

Une première divergence est leur taille : alors que la table 1 actualisée [7] du français comporte quelques dizaines d'entrées seulement, la table correspondante de l'espagnol en contient trois fois plus, p.ex.: [8]

(9) a. Ana (se echó + rompió) a llorar
Ana (se jeta + rompit) à pleurer
'Ana se mit à pleurer'
b. Ana (suele + vuelve a) gritar
Ana (a l'habitude de + retourne) crier
Ana crie (d'habitude + de nouveau)
c. Acertó a pasar por ahí un muchacho
Arriva par hasard à passer par là un garçon
'Un garçon passa par hasard par là '

Il y a non seulement plus de verbes, il y a aussi plus de variation dans les prépositions qui précèdent l'infinitif. Alors que celles-ci correspondent à *Ø*, *à* ou *de* en français, en espagnol on trouve *Ø*, *a*, *de* mais aussi *en*, *con*, *por* et *sin* :

(10) a. La primavera no tardará en venir
Le printemps ne tardera pas dans venir
'Ce sera bientôt le printemps'
b. Amenaza con llover

---

[7] La table 1 actualisée correspond à la table 1 de Gross (1975) sans les locutions verbales figées du type *avoir vite fait de*, *être en voie de*, *être sur le point de*, etc. qui dans le classement actuel correspondent à des verbes composés (Gross : c.p.). La table 1 de l'espagnol ne contient pas de "verbes composés".

[8] Il s'agit de 58 verbes français contre 169 verbes espagnols. Pour des raisons pratiques, tous les verbes qui admettent plus d'une préposition devant l'infinitif ont été comptés comme des entrées différentes. Le même principe ayant prévalu pour les deux langues, la proportion de 1 à 3 reste valable.

Il menace avec pleuvoir
'Il menace de pleuvoir'

c. Max rabiaba por fumar
Max rageait pour fumer
'Max avait très envie de fumer'

d. Max anda sin saber la noticia
Max marche sans savoir la nouvelle
'Max ne connaît toujours pas la nouvelle'

Notons que l'interdiction de la complétive ne tient pas à la préposition. En effet, les prépositions *en* et *con* peuvent introduire des complétives en espagnol :

(11) a. Max insiste en (marcharse + que te marches)
Max insiste dans (partir + que tu partes)
'Max insiste pour (partir + que tu partes)'

b. Max cuenta con (verlo + que lo veas)
Max compte avec (le voir + que tu le voies)
'Max compte (le voir + que tu le voies)'

Par ailleurs, les prépositions *sin* et *por* introduisent des subordonnées adverbiales, qui ne correspondent en aucun cas aux infinitives. En revanche, l'emploi de *por* devant l'infinitif rappelle plutôt, précisément, celui de la préposition devant un SN, comme dans (12c) : [9]

(12) a. * Max anda sin que Ana lo sepa
Max marche sans que Ana le sache

b. * Max rabiaba porque (fumaba + fumara-SUBJ)
Max rageait parce qu'il (fumait + fume)

c. Max rabiaba por un cigarillo
Max rageait pour une cigarette
'Max mourait d'envie d'une cigarette'

Comme je l'ai mentionné au début, une des différences majeures entre les deux langues consiste dans la possibilité qu'ont les verbes de cette classe de se faire suivre d'un SN. Bien que certains cas d'alternance V-inf/SN soient communs aux deux langues, tels *commencer à / par* vs *empezar a / por*, *finir de / par* vs *terminar de / por*, les verbes qui admettent un SN sont bien plus nombreux en espagnol : il s'agit de 95 verbes sur une totalité de 169.

La nature du SN varie selon le cas. Certains SN sont analysables par nominalisation de l'infinitive (13a), d'autres en revanche sont à analyser par suppression d'un verbe support (13b) :

(13) a. Max se demoró mucho en (entregar su artículo + la entrega de su artículo)
Max s'attarda beaucoup dans (remettre son article + la remise de son article)

---

[9] Sur les différences de sens entre les deux prépositions *por* et *para*, voir Bolinger (1944).

'Max mit longtemps à remettre son article'

b. Max no pasará nunca de (botones + ser botones)
Max ne passera jamais de (garçon de courses + être un garçon de courses)
'Max ne sera jamais qu'un garçon de courses'

Quant au traits sémantiques du SN, si la grande majorité des verbes n'admettent qu'un SN de type N-hum, dans certains cas, Nhum est possible également, p.ex. :

(14) Max (enloquecía + se perecía) por (volver a verla + ella)
Max (devenait fou + périssait) pour (la revoir + elle)
'Max mourait d'envie de (la revoir + elle)'

Notons qu'il semble y avoir un rapport entre la possibilité pour ces verbes de sélectionner un SN et le fait de sous-catégoriser un sujet de type Nhum. En effet, parmi les verbes qui admettent un SN, trois quarts sélectionnent un sujet de type exclusivement humain. En revanche, l'absence complète de restrictions de sélection affectant le sujet (propriété notée Nnc dans la table - cf. Annexe) ne vaut que pour une minorité de verbes. Ce sont souvent les mêmes qui n'admettent pas de SN à côté de l'infinitif. Cette complémentarité semble donc pertinente, en particulier pour la question de l'auxiliarité, que je traite dans le point suivant.

En vue des faits observés, on est amené à introduire une distinction importante dans la codification en lexique-grammaire des auxiliaires français et espagnols respectivement. En effet, on est amené à dédoubler la table 1 pour l'espagnol : une table "1a" contiendrait les verbes qui n'admettent pas de SN (et qui ont souvent un sujet non contraint) et une table "1b", quantitativement plus importante, serait réservée aux verbes dont l'infinitif peut alterner avec un SN (et qui ont souvent un sujet Nhum). Du point de vue typologique, la table 1a de l'espagnol présenterait plus de ressemblance avec la table 1 du français que la table 1b.

On voit donc comment la nécessité de disposer de descriptions extrêmement détaillées et exhaustives du matériel linguistique qu'on veut codifier pour des raisons de traitement automatique, permet de révéler des différences entre classes de verbes à première vue très ressemblantes d'une langue à l'autre. Mais j'ai suggéré plus haut que l'existence d'une table 1b en espagnol pourrait également avoir un intérêt d'ordre théorique. Elle pourrait constituer une nouvelle pièce à verser au dossier (déjà compliqué) de l'auxiliarité. Cette classe de verbes représenterait en effet un chaînon supplémentaire dans la chaîne qui va des auxiliaires purs aux verbes lexicaux.

2. Définition de l'auxiliarité.

Les linguistes qui se sont penchés sur la question s'accordent en général à dire, premièrement, que la catégorie des auxiliaires (désormais AUX) existe universellement

(Akmajian et al. 1979, Steele et al. 1981) [10] mais qu'elle est extrêmement difficile à définir (Brieer-Van Akerlaken 1967, De Kock 1975, García 1967, Gómez Torrego 1988, Henrichsen 1967, Ramat 1987, Rojo 1982, Spang-Hanssen 1983, Vikner 1980, Willems 1969). Les critères d'auxiliarité invoqués dans la littérature sont très nombreux et très variés, ils sont d'ordre morphologique, syntaxique ou sémantique.

Critères morphologiques

AUX porte les marques flexionnelles du verbe (temps, nombre, personne)	Brieer-Van Akerlaken 1967, De Kock 1975, Rojo 1982, Willems 1969
AUX a une flexion irrégulière	Ramat 1987, Steele et al. 1981
AUX+V entrent en opposition paradigmatique avec les verbes simples	Brieer-Van Akerlaken 1967, Henrichsen 1967, Vikner 1980, Willems 1969
AUX est un verbe simple (≠ une locution)	Brieer-Van Akerlaken 1967

Critères syntaxiques

Neg et clitiques s'attachent à AUX (AUX+V=1 seul syntagme)	De Kock 1975, Henrichsen 1967, Ruwet 1966, Spang-Hanssen 1983, Vikner 1980
Compléments temporels modifient AUX+V (T_0=T_c)	Brieer-Van Akerlaken 1967, Gross 1975, Spang-Hanssen 1983, Willems 1969
AUX est suivi d'une forme non finie d'un autre verbe	Brieer-Van Akerlaken 1967, Ramat 1987, Rojo 1982

[10] Pour une critique de l'hypothèse contraire ("les Aux sont des verbes comme les autres"), formulée par J. Ross et adoptée par des linguistes comme G. Pullum, voir Akmajian et al. (1979).

AUX est un constituant qui a un comportement syntaxique différent des autres catégories	Akmajian et al. 1979, Steele et al. 1981, Ramat 1987
Dans AUX+V, V ne peut être remplacé par Que P ou SN	Brieer-Van Akerlaken 1967, Gross 1975, Gómez Torrego 1988, Rojo 1982, Ruwet 1966, Willems 1966
AUX n'impose pas de restrictions de sélection (AUX peut exprimer l'impersonnel; AUX est transparent)	Gross 1975, Ramat 1987, Spang-hanssen 1983, Vikner 1980

Critères sémantiques

AUX exprime des éléments notionnels de temps, de mode, d'aspect ou de voix	Akmajian et al. 1979, Brieer-Van Akerlaken 1967, Henrichsen 1967, Spang-Hanssen 1983, Steele et al. 1981, Vikner 1980
AUX a subi une "sublimation" de sens (sens grammatical ≠ sens lexical)	Brieer-Van Akerlaken 1967, Henrichsen 1967, Rojo 1982, Willems 1969
Sens de AUX+V ≠ somme du sens de AUX + sens de V	De Kock 1975, Rojo 1982

Ces critères sont multiples, aucun pourtant n'est entièrement concluant. D'une part, on trouve pour presque tous les critères des contre-exemples. Précisément, beaucoup de verbes espagnols traités ici constituent des contre-exemples du principe syntaxique selon lequel V-inf ne pourrait "être remplacé" par un SN. [11] D'autre part, certains critères valables pour l'ensemble de la

[11] Une approche tout à fait différente consisterait à considérer que c'est le SN qui peut souvent être remplacé par un infinitif en espagnol. L'infinitif aurait donc une valeur nominale (plus poussée qu'en français par exemple), ce qui serait suggéré aussi par le fait qu'il peut, dans certaines conditions, être précédée de l'article. Cette position a eu ses défenseurs, e.a. Lenz (1916). Cette solution a toutefois l'inconvénient de ne pouvoir

classe ne sont pas exclusifs de celle-ci. Ainsi, le critère morphologique selon lequel le premier verbe porte les marques personnelles du verbe vaut de fait pour tous les verbes apparaissant dans une séquence où un verbe conjugué est suivi d'un infinitif. Et enfin, certains critères sont peu opérationnels parce que difficiles à tester. Si la désémantisation du premier verbe (Damourette & Pichon 1911-1936) est évidente dans des cas entièrement grammaticalisés comme *ir* (aller) qui exprime le futur, ou *echar(se)* ((se) jeter) qui indique l'inchoatif, dans bien d'autres, elle est beaucoup plus difficile à calibrer. Soit parce qu'on ne peut dire que le premier verbe se soit entièrement "vidé" de son sens (15a), soit parce qu' on reste très proche du sens compositionnel (15b) :

(15) a. Max revienta por contarlo
Max explose pour le raconter
'Max est très impatient de le raconter'
b. Max se para a pensar
Max s'arrête à penser

Etant donné le caractère peu opérationnel des critères et l'extrême hétérogénéité des verbes surtout, on est forcé d'admettre, comme le font d'ailleurs la plupart des auteurs, qu'il y a des degrés d'auxiliarité. Puisque la grammaticalisation des auxiliaires est un phénomène de diachronie (Berchem 1973, Dietrich 1973), et par conséquent dynamique, il est normal en fait que certains verbes se situent au point d'aboutissement du procès, tandis que d'autres se trouvent à l'autre extrême, entamant à peine le même processus (Lamiroy 1987). Parmi les verbes espagnols décrits ici, beaucoup appartiennent à la dernière catégorie. Toutefois, indépendamment du degré plus ou moins avancé de grammaticalisation, ils présentent tous une propriété formelle commune, à savoir l'absence de complétive. Celle-ci constitue le point de rupture entre la table 1 des (semi)-auxiliaires et les autres tables des verbes.

Selon la définition "universelle" de la catégorie AUX de Akmajian et al. (1979) et Steele et al. (1981), AUX est 1º un constituant 2º qui se distingue par son comportement syntaxique des autres catégories et 3º qui exprime des éléments notionnels de temps, d'aspect ou de modalité. A condition d'admettre que certains verbes sont de meilleurs représentants de la catégorie que d'autres, on pourrait considérer tous les verbes de la table 1 comme des membres de la

---

rendre compte, évidemment, des verbes pour lesquels l'alternance V-inf/SN n'existe pas et de séparer ainsi des verbes qui par ailleurs sont syntaxiquement et sémantiquement proches. Une autre analyse encore (la solution "Perlmutter") consisterait à dire que pour tous les cas où V-inf et SN sont admis, nous avons chaque fois affaire à deux verbes homonymes. Cette solution, qui a le désavantage de présenter le phénomène de l'alterance V-inf/SN comme un accident, serait d'autant plus gênante ici que les verbes "homonymiques" sont extrêmement nombreux.

catégorie AUX. L'intérêt des verbes espagnols dont il a été question ici est alors de montrer que cette représentativité par rapport à la catégorie AUX est associable à des propriétés formelles de sélection : en l'occurrence, les verbes de la table "1b", qui admettent un SN, sont moins représentatifs de la catégorie AUX que ceux de la table "1a", qui excluent le SN. [12] Prise globalement, la table 1 du français est plus représentative de AUX que la table 1 de l'espagnol. On pourrait schématiser comme suit :

AUX+V		V+QUE P
V-inf	V-inf	que P
* SN	SN	SN
* que P	* que P	V-inf

<----------

GRAMMATICALISATION

GRAMMAIRE LEXIQUE

Une des conclusions théoriques auxquelles on arrive donc est que du point de vue typologique, la grammaticalisation semble plus avancée en français qu'en espagnol : la table 1 du français est plus résiduelle dans la mesure où elle contient moins de verbes, qui à leur tour présentent plus de symptômes de grammaticalisation que les verbes correspondants en espagnol, en particulier l'absence de SN. [13] Mais cette conclusion n'est pas tout à fait surprenante après tout : elle confirme ce qui est suggéré par d'autres domaines de la langue, tels la phonétique ou la morphologie (Lamiroy 1993), où l'observation de données indépendantes converge vers le même résultat, à savoir que les langues romanes n'évoluent pas toutes à la même vitesse et que dans l'ensemble, c'est d'habitude - sans qu'on sache pourquoi - le français qui est en tête.

---

[12] De même, les verbes pour lesquels N_0 = exclusivement Nhum sont moins représentatifs que ceux qui ont Nnc comme sujet.

[13] Le fait que les verbes de mouvement prennent exclusivement un infinitif en français, alors qu'ils admettent aussi la complétive correspondante en espagnol, pourrait relever du même phénomène.

Références

AKMAJIAN, A., S. STEELE & T. WASOW. 1979.

The Category AUX in Universal Grammar. *Linguistic Inquiry* , 10, 1-64.

BERCHEM, T. 1973.

Studien zum Funktionswandel bei Auxiliaren und Semi-Auxiliaren in den romanischen Sprachen. Tübingen : Niemeyer.

BOLINGER, D. 1944.

Purpose with por and para. *The Modern Language Journal*, 28, 15-21.

BOONS, J.P., A. GUILLET & C. LECLÈRE. 1976.

La structure de la phrase simple en français. Constructions intransitives . Genève : Droz.

BRIEER-VAN AKERLAKEN. 1967.

Le problème des verbes auxiliaires en français contemporain. *Folia Linguistica* , 1, 194-231.

CLOSS TRAUGOTT, E. & HEINE, B.1991.

Approaches to grammaticalisation . Amsterdam : Benjamins.

DAMOURETTE, J. & E. PICHON. 1911-1936.

Des Mots à la Pensée. Essai de Grammaire de la Langue française. Paris : d'Artrey, vol. III.

DE KOCK, J. 1975.

Pour une nouvelle définition de la notion d'auxiliarité. *La linguistique* , II, 2, 81-92.

DIETRICH, W. 1973.

Der periphrastische Verbalaspekt in den romanischen Sprachen. Tübingen : Niemeyer.

GARCIA, E. 1967.

Auxiliaries and the criterion of simplicity. *Language*, 43, 853-870.

GOMEZ TORREGO, L. 1988.

Perífrasis verbales. Madrid : Arco/Libris.

GREVISSE-GOOSSE, A. 1993.

Le Bon Usage . Gembloux : Duculot.

GROSS, M. 1975.

Méthodes en Syntaxe. Paris : Hermann.

GUILLET, A. & C. LECLERE. 1992.

La structure des phrases simples en français. Constructions transitives locatives . Genève : Droz.

HENRICHSEN, A.J. 1967.

Les périphrases verbales du français moderne. *Revue Romane*, numéro spécial 1, 45-56.

LAMIROY, B. 1987.

The Complementation of Aspectual Verbs in French. *Language*, 63, 2, 278-298.

LAMIROY, B. 1991.

Léxico y Gramática del Español. Estructuras Verbales de Espacio y Tiempo. Barcelona: Anthropos.

LAMIROY, B. 1993.

La dichotomie synchronie-diachronie et la typologie des langues romanes. In : W. Raible & W. Osterreicher (éds.). *Actes du XXe Congrès International de Linguistique et Philologie Romanes* . Munich : Saur, III, IV, 211-221.

LEMHAGEN, G. 1979.

La concurrence entre l'infinitif et la subordonnée complétive introduite par que en français contemporain. Upsala : Acta Universitatis Upsaliensis.

LENZ, R. 1916.

La oración y sus partes . Santiago de Chile.

PERLMUTTER, D. 1970.

The two verbs *begin*. In : R. Jacobs & P. Rosenbaum (éds.). *Readings in English Transformational Grammar* . Waltham, MA : Ginn, 107-120.

RAMAT, P. 1987.

Introductory Paper. In : M. Harris & P. Ramat (éds.) *Historical Development of Auxiliaries* . Berlin : Mouton-de Gruyter, 3-19.

ROJO, G. 1982.

Aportaciones al estudio de la auxiliaridad. *Actas del Cuarto Congreso Internacional de Hispanistas* . Salamanca : Universidad de Salamanca, 499-508.

RUWET, N. 1966.

Le constituant "Auxiliaire" en français moderne. *Langages*, 4, 105-121.

RUWET, N. 1983.

Montée et Contrôle : une question à revoir ? *Revue Romane*, numéro spécial 24, 17-37.

SKYDSGAARD, S. 1977.

La combinatoria sintáctica del infinitivo español. Madrid : Castalia.

SPANG-HANSSEN, E. 1983.

La notion de verbe auxiliaire. *Revue Romane* , numéro spécial 24, 6-16.

STEELE, S. ET AL. 1981.

An Encyclopedia of AUX : a study in cross-linguistic equivalence . Cambridge : MIT Press.

SUBIRATS, C. 1987.

Sentential Complementation in Spanish . Amsterdam : Benjamins.

VIKNER, C. 1980.

Linfinitif et le syntagme infinitif. *Revue Romane*, 15, 2, 252-291.

WILLEMS, D. 1969.

Analyse des critères d'auxiliarité. *Travaux de linguistique* , 1, 87-99.

ANNEXE :
Extrait de la table 1 de l'espagnol

N_0				V-inf	Que	Prep N_1		N_1		
Nhum	Nnc			W	P	Nhum	N-hum	Nhum	N-hum	
+	-	batallar	por	+	-	-	+	-	-	[se démener]
+	+	caracterizarse	por	+	-	-	+	-	-	[se caractériser]
+	-	ceñirse	a	+	-	-	+	-	-	[se limiter]
+	+	cesar	de	+	-	-	-	-	+	[cesser]
+	-	chiflarse	por	+	-	+	+	-	-	[raffoler]
+	-	circunscribirse	a	+	-	-	+	-	-	[se limiter]
+	+	comenzar	a	+	-	-	-	+	-	[commencer]
+	+	comenzar	por	+	-	+	+	-	-	[commencer]
+	-	concretarse	a	+	-	-	+	-	-	[se borner]
+	-	contenerse	de	+	-	-	-	-	-	[se retenir]
+	+	continuar	sin	+	-	-	+	-	-	[continuer]
+	-	dar	a	+	-	-	-	-	-	[se mettre]
+	-	dar	en	+	-	-	-	-	-	[se mettre]
+	+	dar	por	+	-	-	+	-	-	[se mettre]
+	-	darse	a	+	-	-	-	-	-	[se mettre]
+	-	darse	en	+	-	-	-	-	-	[se mettre]
+	+	deber	∅	+	-	-	-	-	-	[devoir]
+	+	deber	de	+	-	-	-	-	-	[devoir]
+	-	decidirse	por	+	-	+	+	-	-	[se décider]
+	-	declinar	∅	+	-	-	-	-	+	[refuser]
+	+	dejar	de	+	-	-	-	-	+	[cesser]
+	-	demorarse	en	+	-	-	+	-	-	[s'attarder]
+	-	descararse	a	+	-	-	-	-	-	[se gêner]
+	-	descuidarse	de	+	-	+	+	-	-	[omettre]
+	-	desdeñar	∅	+	-	-	-	-	+	[dédaigner]
+	-	desdeñarse	de	+	-	-	-	-	-	[dédaigner]
+	-	desesperarse	por	+	-	+	+	-	-	[désespérer]
+	-	desgañitarse	por	+	-	-	+	-	-	[s'éreinter]
+	-	deshacerse	por	+	-	-	+	-	-	[se démener]
+	-	desistir	de	+	-	-	+	-	-	[se désister]

Experiences in Lexical Disambiguation Using Local Grammars

ÉRIC LAPORTE

Abstract

Lexical disambiguation is one of the major challenges facing those who devise lexical taggers. Choosing a methodology for lexical disambiguation implies the following: defining a tag set, deciding whether one or several solutions are retained in disambiguated text, deciding whether lexical disambiguation is performed after an initial tagging, and choosing between handcrafted and statistical data. We present the choices made in two recent contributions, Silberztein (1989, 1993) and Roche (1992). We give a formal description of the two tag sets and of the two methods. We show that none of these methods is formally more powerful than the other. We also compare them from a practical point of view.

1. Introduction

Many words are ambiguous in their part of speech. For example, *show* can be a noun or a verb. However, when a word appears in a text, the ambiguity is usually much reduced: in *Several of the most important studies show that experience is critical*, the word *show* can only be a verb. A lexical tagger is a system that assigns lexical categories to words. Disambiguating of lexical categories consists in using context to reduce the number of lexical categories assigned to words. Disambiguation of lexical categories is one of the major challenges facing those who devise lexical taggers.

The problem of tagging words with lexical categories arises quite often in natural language processing, e.g. in spelling correction, grammar and style checking, phrase recognition, text-to-speech conversion, corpus analysis... In parsing, the correct tagging of words is a side effect of parsing[1], but if lexical categories are previously assigned to words, higher-level analysis is usually facilitated (Milne, 1986; Hindle, 1989; Rimon, Herz, 1991; Cutting et al., 1992). Large disambiguated text corpora are important resources for many applications, e.g. for training probabilistic systems, and manual tagging is slow, expensive and error-prone. Thus, lexical taggers are useful as a front end to many natural language processing systems.

2. Methodology

The tag set

A lexical tagger labels each word-form in a text with one or several tags, i.e. codes which convey lexical information. When this information consists of part of speech only, there are about 10 to 20 lexical categories. If it includes some more grammatical data, the lexical categories are finer and more numerous. These grammatical data may be:

- lemma, e.g. *show* for the word-form *shown*, or information needed to obtain the lemma from the word-form;
- inflectional features (gender, tense...), which are a basic information in Romance languages;
- delimitation of compound words, i.e. frozen sequences of at least two simple words separated by graphic separators (e.g. *of course, text processor*), in which case specific tags are assigned to compounds, e.g. *Adverb* for *of course*;

If lemmata are explicitly given in tags, and if inflectional codes, e.g. conjugation codes, are also included in tags, word-forms can be deduced from their tags. Higher-level information, e.g. syntactic relation to the predicate (Koskenniemi, 1990), is seldom considered in disambiguation of lexical categories.

The tagged Brown Corpus uses a set of 87 simple tags (Garside, Leech, Sampson, 1987, pp. 165-183) which has been used later in other projects. For French, a tag set with part of speech and inflectional features only has approximately the same size. This paper describes experiences in French with lexical categories including lemma in lexical information, both for simple words and for compounds; the word-form can be deduced from the tag. The size of the tag set is thus that of a lexicon of the language. In the following we assume that the word-form can be deduced from the tag.

The form of disambiguated text

The output of most lexical taggers for an input text is a sequence of word/tag pairs: a unique tag is assigned to each word. This choice may a priori be supported by two assertions: (i) that it is possible to build or tune a lexical tagger so as to assign a unique tag to each word without any error; or (ii) that it is not a serious imperfection for a lexical tagger to assign a unique wrong tag to a word in a text. The status of assertion (i) is unclear, unless we consider a parser as a part of or a front end to the lexical tagger instead of the reverse: even with a coarse tag set, the correct lexical tagging of some natural sentences involves recognizing the syntactic structure of the whole sentence or even understanding its meaning. Assertion (ii) is controversial too, especially in the context of parsing: a manual correction of the output of the lexical tagger is sometimes impracticable; it is natural for a parser to rule out analyses, but not to create new analyses with parts of speech different from those in the input. Moreover, if a unique tag is assigned to each word, even actually ambiguous sentences are represented as lexically unambiguous.

In contrast, a number of recent lexical taggers

[1] This is considered as the origin of the verb *parse*.

(Silberztein, 1989; Koskenniemi, 1990; Rimon, Herz, 1991; Roche, 1992) allow for several solutions when the text is lexically ambiguous and when the only right solution cannot be found. These contributions aim at guaranteeing zero silence: the correct tag(s) for a word should never be ruled out. In other words, an analysis can be discarded only if there is no doubt that it is wrong. This objective is not often mentioned, and unrealistic for taggers that assign a unique tag to each word, unless assertion (i) is assumed.

Since tags of words in the same sentence are not independent, the output of such systems for a given sequence of words is a set of one or several sequences of tags. The set of tag sequences for a given input sequence is represented in an appropriate form. Since it is a finite set of sequences which usually have much in common, this form is always that of an acyclic finite-state automaton, a notion also termed as directed acyclic graph (DAG) or directed acyclic word graph (DAWG), finite-state machine (FSM) or finite-state network (Koskenniemi, 1990), sentence graph (Rimon, Herz, 1991), or word lattice (Vosse, 1992). One of the advantages of using acyclic automata in this context is that it systematizes the representation of lexical ambiguity, both before and after disambiguation, and for all types: parts of speech, inflectional features, phrase ambiguities (compound vs. sequence of simple words). Fig. 1 exemplifies part of speech ambiguity for *traverse* "to cross"; "strut", and phrase ambiguity for *chemin de fer* "railway" which contains *chemin* "path" and *fer* "iron".

Initial tagging and lexical disambiguation

Most lexical taggers divide the task into two steps: first (initial tagging), word forms are considered out of context to make out the list of all tags for each word; second (lexical disambiguation), context is taken into account in order to select a subset of the initial tagged sequences.

In other lexical taggers (Klein, Simmons, 1963; Dermatas, Kokkinakis, 1989; Pelillo, Refice, 1991; Brill, 1992; Federici, Pirrelli, 1992), both subtasks are carried out at the same time and a disambiguated output is built directly, generally in order to avoid the construction of a large dictionary.

The modular approach seems to us a reasonable one. The two subtasks are clearly defined. Once a tag set has been chosen, the subtasks are independent: one can expect that so could be the methods to achieve them with the best results. Improving initial tagging is a matter of morphological description of words, whereas refining upon lexical disambiguation is a matter of grammatical description of word sequences.

This approach is coherent with the use of acyclic automata for the representation of lexical ambiguity. After the initial tagging, the set of sequences recognized by the automaton is the set of a priori possible tag sequences for the input sequence. During lexical disambiguation, the automaton is modified. The number of sequences recognized by the automaton decreases, but the number of states and transitions in the automaton may increase or decrease.

The modular approach is of course of a particular interest when one uses a reliable morphological dictionary that gives for each word form, either simple or compound, the list of possible tags. In this case, initial tagging is simply performed by dictionary lookup. Such a framework for French was developed at the LADL and the CERIL with the lexicons DELAF (Courtois, 1990) and DELACF (Silberztein, 1990), and with the compression and lookup algorithms implemented by Revuz (1991) and Roche (1992) to improve upon tries (Knuth, 1973). It is now integrated into the system INTEX (Silberztein, 1993). The compressed size of DELAF is less than 900 Ko for 700.000 forms.

Handcrafted vs. statistical data

The information used by lexical taggers in order

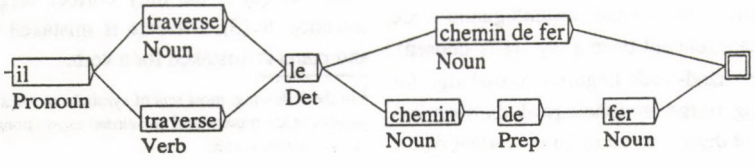


Fig. 1

to disambiguate words may be hand-made grammatical knowledge or statistical data learnt automatically in large text corpora. The tagger in Hindle (1989) uses a mixture of both. Examples of lexical taggers with hand-made grammatical knowledge are Klein, Simmons (1963), Hindle (1983), Silberztein (1989), Paulussen, Martin (1992), Roche (1992). In Rimon, Herz (1991), the data are automatically produced from hand-made context-free grammars. In all of these contributions, the grammatical data are represented in finite-state automata, or could easily be. Examples of lexical taggers with statistical data are Greene, Rubin (1971), Hindle (1989), Brill (1992), Federici, Pirrelli (1992), where the data are rules produced through statistical processes; Marshall (1983), Jelinek (1985), DeRose (1988), Cutting et al. (1992), etc., where the data consist of tables of statistics, e.g. parameters of a Markov model. All of these contributions assign a unique tag to each word.

The arguments in favour of any of these two types of approaches often refer to their ability to contend with the diversity of unrestricted text. Some doubt that hand-made linguistic knowledge can take into account all that can happen in unrestricted text. Others think that information learnt automatically in a text corpus, even in a large one with carefully distributed samples of text, is not accurate enough for unrestricted text. It seems to us that the use of hand-written linguistic knowledge is better adapted if one aims at zero silence, i.e. at guaranteeing that an analysis is discarded only if there is no doubt that it is wrong. The writer of the linguistic data can use a text corpus as an aid, but must be able to create counter-examples that are not in the corpus, so that the resulting linguistic data may be independent of it.

We will focus on lexical tagging with the objective of zero silence. A method adapted to this objective combines the possibility of several solutions (the output for an input word sequence is an acyclic automaton), the modular approach (initial tagging and lexical disambiguation are considered independent once a tag set is chosen), and the use of hand-made linguistic knowledge for initial tagging (large morphological dictionaries) and for lexical disambiguation (grammatical data). The only contributions in this framework are those of Silberztein (1989, 1993) and Roche (1993). Both consist of implemented algorithms and use the same dictionaries. In the following, we make formal descriptions of the two systems and a formal and practical comparison. Mathematical terminology is that of Berstel (1979) with the exception that the empty word is denoted by ε.

3. The tag sets

Roche (1992)

In this system, a tag is a sequence of several symbols that belong to a finite alphabet A_1, which is defined as follows. Let

$Cat = N \mid V \mid A \mid ADV \mid DET \mid DETP \mid DETQ \mid PRO \mid PREP \mid CONJC \mid CONJS \mid INTJ \mid XINC$

be the set of parts of speech[2],

$Mf = N1 \mid N2 \mid N3 \mid ... \mid V1 \mid V2 \mid V3 \mid ... \mid A1 \mid A2 \mid A3 \mid ... \mid A79 \mid A80$

be the set of inflectional codes, e.g. conjugation codes,

$Voc = a \mid à \mid aa \mid aabam \mid aalénien \mid aalénienne \mid aaléniennes \mid aaléniens \mid ... \mid zyzomys \mid zzz$

be the set of all possible inflected forms in the language, including compounds like *chemin de fer*,

$Val = v\text{-}t \mid P \mid F \mid C \mid I \mid J \mid S \mid T \mid Y \mid G \mid K \mid W \mid 1 \mid 2 \mid 3 \mid m \mid f \mid s \mid p$

be the set of inflectional feature values, and *Unit* be a symbol which appears in front of every token. Now $A_1 = Unit \mid Cat \mid Mf \mid Voc \mid Val$; the tag set is

$C \subset Unit\ Cat\ (Mf\ Voc^+\ Val^+ \mid \varepsilon)\ \ Voc^+ \subset {A_1}^+$

C is a code because the first symbol of every element of C is the symbol *Unit*.

Two of the initial taggings of

(1) *Il traverse un cours d'eau* "He crosses a river"

take the following forms:

(2) *Unit PRO il m s 3 Il Unit V 3 traverser P 3 s traverse Unit DET un m s un Unit N cours/d'eau m s cours/d'eau*

(3) *Unit PRO il m s 3 Il Unit N 21 traverse f s traverse Unit DET un m s un Unit V 31 courir P 2 s cours Unit PREP d Unit N 23 eau f s eau*

Note that (2) is the only correct tagging of the sentence. In (3), *traverse* is mistaken for a noun and *cours* is mistaken for a verb.

[2] In the following, most sets of symbols and sets of sequences are represented as rational expressions, and the bar | stands for set union.

We define a rational word function

$\varphi_1 : A_1^* \dashrightarrow Voc^*$

which maps any sequence of tags into the corresponding sequence of word-forms by deleting all other information; $\mathrm{dom}\varphi_1 = C^*$. Examples:

φ_1(*Unit DET le Unit N N1 passe m s passe*) = *le passe*

φ_1^{-1}({*le passe*}) = *Unit* (*DET* | *PRO*) *le Unit* (*N* (*N1 passe m* | *N21 passe f*) | *V V3 passer v-t* ((*P* | *S*)(1 | 3) | *Y2*)) *s passe*

φ_1(*Unit N;NA coup fumant m s coup fumant*) = *coup fumant*

The result of dictionary lookup for an input sequence t in Voc^* is $\varphi_1^{-1}(\{t\})$. This set of tags is represented in the form of an acyclic automaton on the alphabet A_1.

Silberztein (1989, 1993)

It uses another alphabet, A_3, which contains all possible complete word tags in the lexicon, e.g.:

<il,PRO:ms3>
<traverser,V3:P3s>
<cours/d'eau,NDN:ms>

The tag gives the lemma, the part of speech, the inflectional code if any, and after a semicolon the sequence of inflectional features if any. The a priori sentence tags (2) and (3) above now take other forms:

(4) *<il,PRO:ms3> <traverser,V3:P3s> <un,DET:ms> <cours/d'eau,NDN:ms>*

(5) *<il,PRO:ms3> <traverse,N21:fs> <un,DET:ms> <courir,V31:P2s> <de,PREP> <eau,N23:fs>*

We define a continuous morphism

$\varphi_2 : A_3^* \dashrightarrow Voc^*$

which maps any sequence of tags into the corresponding sequence of word-forms by deducing the word-form and deleting all other information. Examples:

φ_2(*<le DET:ms> <passe N1:ms>*) = *le passe*

φ_2^{-1}({*le passe*}) = (*<le DET:ms>* | *<le PRO:ms3s>*)(*<passer V3:P1s>* | *<passer V3:P3s>* | ... | *<passe N21:fs>*)

φ_2(*<coup/fumant N;NA:ms>*) = *coup fumant*

The result of dictionary lookup for an input sequence t in Voc^* is $\varphi_2^{-1}(\{t\})$. This set of tags is represented in the form of an acyclic automaton on the alphabet A_2.

The elements of A_3 are completely specified tags. This system also uses another alphabet, A_2, which contains incomplete grammatical specifications, namely:

- all possible simple word-forms, i.e.

Voc = *a* | *à* | *aa* | *aabam* | *aalénien* | *aalénienne* | *aaléniennes* | *aaléniens* | ... | *zyzomys* | *zzz*

- incomplete grammatical specifications on part of speech, inflectional feature values, lemma, or a combination of these, e.g.:

<ADV> for any adverb
<V> for any verb form
<V:Ps> for any verb form in the present singular
<traverser> for any word-form whose lemma is *traverser*
<passeur, N:s> for any singular form of the noun passeur

<prendre:P>	*<V:P>*
<prendre:P3p>	*<V:P3p>*
<prendre:p>	*<V:p>*

- the symbol *<MOT>* which matches any simple word.

If and only if an element a of A_3 may satisfy the constraints expressed in an element b of A_2, we write $a \in \sigma(b)$, thus defining a substitution σ from A_2^* into the power-set of A_3^*:

σ(*<prendre:P3p>*) = *<prendre V66:P3p>*

σ(*<prendre:P>*) = *<prendre V66:P1s>* | *<prendre V66:P2s>* | ... | *<prendre V66:P3p>*

σ(*passe*) = *<passer V3:P1s>* | *<passer V3:P3s>* | ... | *<passe N21:fs>*

σ(*<MOT>*) = σ(*Voc*)

Relations between A_1 and A_3

The tag sets C and A_3 are equivalent. We define an injective morphism

$\alpha : A_3^* \dashrightarrow A_1^*$

and we extend it to an isomorphism from A_3^* onto C^*. Example:

α(*<le DET:ms> <passe N1:ms>*) = *Unit DET le Unit N N1 passe m s passe*

We have $\varphi_1 \circ \alpha = \varphi_2$.

4. Specifying a formal language defines a lexical disambiguation

Let $L \subset C^*$ be a set of tag sequences: for each text $t \in Voc^*$, the set $\varphi_1^{-1}(\{t\}) \cap L$ is the set of taggings of t that are in L. Let *Corr* be the set of

correct taggings of all texts: $\varphi_1^{-1}(\{t\}) \cap Corr$ is the set of correct taggings of t. If we specify L and compute $\varphi_1^{-1}(\{t\}) \cap L$, the noise in the tagging is thus

$$\varphi_1^{-1}(\{t\}) \cap L \setminus Corr$$

and the silence is

$$\varphi_1^{-1}(\{t\}) \cap Corr \setminus L$$

To ensure that silence rate is zero, we have to specify L so that each correct text tagging be in L. We call the specification of L a local grammar, since it deals with grammatical notions and it can be robust for local constraints but not, up to now, for global constraints in sentences. Usually, L is a rational language.

The same local grammar can be used to tag texts and to detect errors. Whenever $\varphi_1^{-1}(\{t\}) \cap L$ is empty, the text t has no correct tagging and contains an error.

Roche (1992)

Let $F \subset A_1^*$ be a set of sequences that can never appear in a correct text tagging, e.g.

$$Unit\ DET\ un\ m\ s\ un\ Unit\ V\ A_1\ A_1\ P$$

where A_1 stands for any element of A_1. This interdiction means that the determiner *un* cannot be followed by a verb in the present. The set

$$R(F) = C^* \setminus A_1^* F A_1^*$$

is the set of taggings that have no factor in the set F of forbidden sequences: every correct text tagging should be in $R(F)$. A lexical disambiguation is thus defined by a specification of $R(F)$. This specification takes the form of a finite-state automaton which recognizes F and is called a local grammar. A lexical disambiguation is performed by selecting, for each text $t \in Voc^*$ and each rational set F, the set $\varphi_1^{-1}(\{t\}) \cap R(F)$ of taggings of t with no forbidden sequences. The efficient algorithm of Roche (1992) constructs an automaton that recognizes $\varphi_1^{-1}(\{t\}) \cap R(F)$ directly from an automaton that recognizes the set $\varphi_1^{-1}(\{t\})$ and an automaton that recognizes F.

This type of local grammars are cumulative: for each rational sets $F_1, F_2 \subset A_1^*$,

$$R(F_1 \cup F_2) = R(F_1) \cap R(F_2)$$

For each rational language F, there is a rational language F_0, minimal for set inclusion, such that:

$$R(F_0) = R(F)$$

and the elements of F_0 are minimal in F and in F_0 for factor ordering.

Silberztein (1989, 1993)

This algorithm is exemplified and informally described in Silberztein (1993). We give a more formal specification of the lexical disambiguation performed by this algorithm.

As above, let $L \subset A_3^*$ be a set of tag sequences: for each text $t \in Voc^*$, the set $\varphi_2^{-1}(\{t\}) \cap L$ is the set of taggings of t that are in L. If each correct text tagging is in L, computing $\varphi_2^{-1}(\{t\}) \cap L$ performs a lexical disambiguation.

The local grammar is a strictly alphabetic finite transducer, i.e. a finite automaton each transition of which is labelled by one element in A_2 as input label and one element in A_2 as output label. It

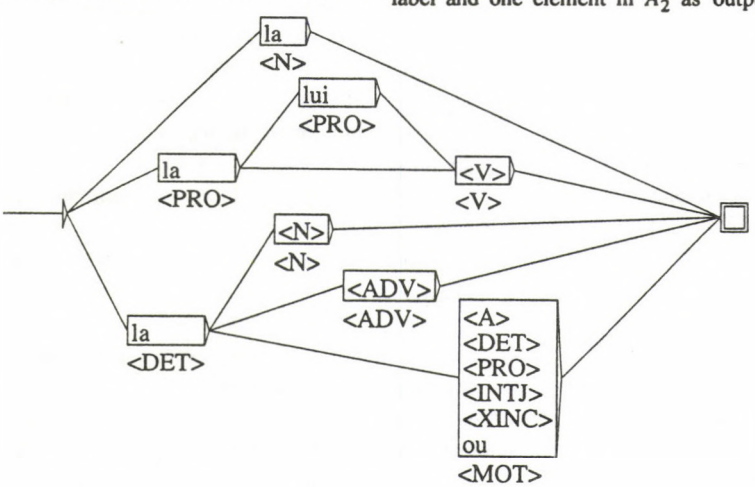


Fig. 2.

realizes a rational transduction $\tau \in \mathrm{Rat}(A_2{}^* \times A_2{}^*)$, i.e. a relation between input sequences in $A_2{}^*$ and output sequences in A_2. The transduction must have the property that $(\varepsilon, \varepsilon) \notin \tau$ where ε is the empty word. Fig. 2 is an example of this type of local grammar. The set $\mathrm{dom}\tau$ is the set of possible input sequences of τ and will recognize grammatical sequences. The set $\mathrm{im}\tau$ is the set of possible output sequences of τ and will impose grammatical constraints on the sequences recognized.

Let ρ be the equivalence relation over $A_3{}^*$ defined by:

$$\forall u, v \in A_3{}^* \quad (u \rho v \Leftrightarrow$$
$$(|u| = |v| \text{ and } \forall i \in [1, |u|] \ \varphi_2(u_i) = \varphi_2(v_i)))$$

e.g. *<superbe N21:fs> <gaulliste A31:fs>* and *<superbe A31:fs> <gaulliste N31:fs>* are in relation by ρ but *<pomme/de/terre N;NDN:fs> <cuire V91:Kfs>* and *<pomme N21:fs> <de PREP> <terre/cuite N;NA:fs>* are not.

We define a transduction $g(\tau) \in \mathrm{Rat}(A_3{}^* \times A_3{}^*)$ by:

$$\forall u, v \in A_3{}^* \quad ((u, v) \in g(\tau) \Leftrightarrow$$
$$(\exists (u', v') \in \tau \ (u \in \sigma(u') \text{ and } v \in \sigma(v')) \text{ and } u \rho v))$$

For each text $t \in Voc^*$, we define a transduction $f(t, \tau) \in \mathrm{Rat}(A_3{}^* \times A_3{}^*)$ by:

$$\forall u, v \in A_3{}^* \quad ((u, v) \in f(t, \tau) \Leftrightarrow$$
$$(u, v \in \mathrm{Pref}(\varphi_2{}^{-1}(\{t\})) \text{ and } (u, v) \in g(\tau))$$

For each text t, we define a set $F_\tau(t) \subset A_3{}^*$ by:

if $\sigma(\mathrm{dom}\tau) \cap \mathrm{Pref}(\varphi_2{}^{-1}(\{t\})) = \emptyset$,
$$F_\tau(t) = A_3 \cap \mathrm{Pref}(\varphi_2{}^{-1}(\{t\}));$$
else $F_\tau(t) = \mathrm{im} f(t, \tau)$.

Now we define a set $S(\tau) \subset A_3{}^*$ by defining the set $S(\tau) \cap \varphi_2{}^{-1}(\{t\})$ for each t by induction on $|t|$:

$S(\tau) \cap \varepsilon = \varepsilon$;
if $|t| > 0, S(\tau) \cap \varphi_2{}^{-1}(\{t\})$
$$= \bigcup_{u \in F_\tau(t)} u.(S(\tau) \cap \varphi_2{}^{-1}(\{\varphi_2(u)^{-1} t\}))$$
$$= \{uv \in A_3{}^* \mid \varphi_2(u)\varphi_2(v) = t \text{ and } u \in F_\tau(t) \text{ and } v \in S(\tau)\}$$

An equivalent definition of $S(\tau)$ uses a new symbol $@ \notin A_2 \mid A_3$ and the projection

$$\varphi_@ : (A_3 \mid @)^* \dashrightarrow A_3{}^*$$

such that $\varphi_@(@) = \varepsilon$:

$$S(\tau) = \varphi_@((@A_3 \mid \mathrm{im}g(\tau))^* \setminus$$
$$(@ \mid A_3)^* @ \varphi_@{}^{-1}(\varphi_2{}^{-1}(\varphi_2(\sigma(\mathrm{dom}\tau))Voc^*))$$

The local grammar is written so that each correct text tagging be in $S(\tau)$. A function of INTEX (Silberztein, 1993) constructs an automaton that recognizes $\varphi_2{}^{-1}(\{t\}) \cap S(\tau)$ directly from an automaton that recognizes the set $\varphi_2{}^{-1}(\{t\})$ and a transducer that realizes τ. For example, the local grammar in Fig. 2 applied to the text *la veine porte* produces the automaton in Fig. 3.

5. A formal comparison

From a formal point of view, both methods provide possibilities of defining rational languages, but these theoretical possibilities are not exactly the same for the two methods, i.e. there exist some $S(\tau)$ that cannot be expressed in the form $\alpha^{-1}(R(F))$, and some $R(F)$ that cannot be expressed in the form $\alpha(S(\tau))$.

An $S(\tau)$ which is not an $\alpha^{-1}(R(F))$

For each rational language F, the following holds:

$$\forall u_1, u_2 \in A_1{}^* \$$
$$(u_1 u_2 \in R(F) \Rightarrow u_1 \in R(F) \text{ and } u_2 \in R(F))$$

Now define τ with the graph of Fig. 4. Assume that there is a rational language $F \subset A_1{}^*$ such that $S(\tau) = \alpha^{-1}(R(F))$. Then

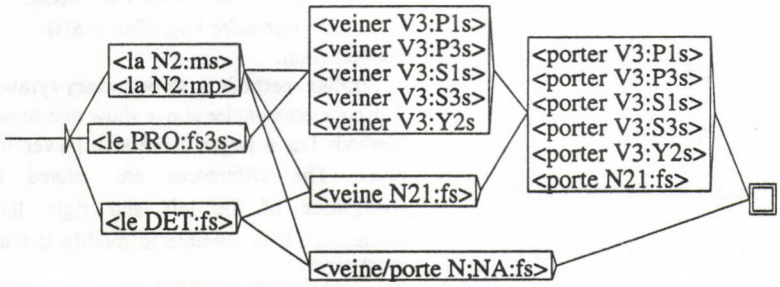
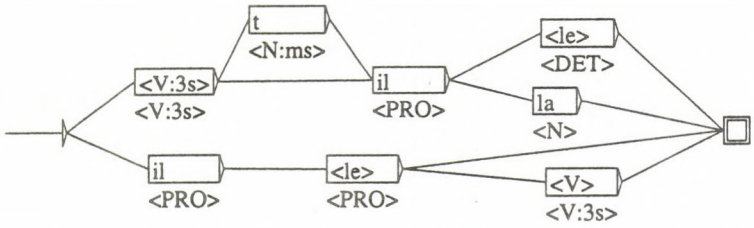


Fig. 3.

Fig. 4.



$\forall\, u_1, u_2 \in A_3^*$

$(u_1u_2 \in S(\tau) \Rightarrow u_1 \in S(\tau)$ and $u_2 \in S(\tau))$

In fact, with $\varphi_2(u_1)$ = *sent* and $\varphi_2(u_2)$ = *il la touche*:

u_1u_2 = <*sentir V28:P3s*> <*il PRO:ms3s*> <*le DET:fs*> <*touche N21:fs*> $\in S(\tau)$

u_1 = <*sentir V28:P3s*> $\in S(\tau)$

but

u_2 = <*il PRO:ms3s*> <*le DET:fs*> <*touche N21:fs*> $\notin S(\tau)$

An $R(F)$ which is not an $\alpha(S(\tau))$

Some $R(F)$ cannot be expressed in the form $\alpha(S(\tau))$. It is the case whenever all elements of F are minimal for factor ordering, f and g are elements of $\alpha^{-1}(F) \setminus \varepsilon$, and f has a prefix $h \neq f$ such that $\varphi_2(h) = \varphi_2(g)$. Then no transduction τ in Rat($A_2^* \times A_2^*$) satisfies $R(F) = \alpha(S(\tau))$.

We prove it for a particular language:

F = Unit PRO il Unit DET le | Unit PRO il Unit PRO le Unit V V66 prendre v-t P 2 s prends

$\alpha^{-1}(F)$ = <*il PRO:ms3s*> <*le DET:ms*> | <*il PRO:ms3s*> <*le PRO:ms3s*> <*prendre V66:P2s*>

Assume there exists $\tau \in$ Rat($A_2^* \times A_2^*$) such that $R(F) = \alpha(S(\tau))$.

Assume that $\sigma(\text{dom}\tau) \cap \text{Pref}(\varphi_2^{-1}(\{il\ le\})) = \varnothing$.

Then

$F_\tau(il\ le)$ = <*il PRO:ms3s*>

$\varphi_2^{-1}(\{il\ le\}) \cap S(\tau) =$

<*il PRO:ms3s*>$(\varphi_2^{-1}(\{le\}) \cap S(\tau))$

but

Unit DET le $\in R(F)$

Consequently,

<*le DET:ms*> $\in S(\tau)$

and

<*il PRO:ms3s*> <*le DET:ms*> $\in S(\tau)$

whereas

Unit PRO il Unit DET le $\notin R(F)$:

contradiction.

Thus $\sigma(\text{dom}\tau) \cap \text{Pref}(\varphi_2^{-1}(\{il\ le\})) \neq \varnothing$ and

$F_\tau(il\ le) = \text{im}f(il\ le, \tau)$.

We have $\varphi_2^{-1}(\{il\ le\}) \subset \text{Pref}(\varphi_2^{-1}(\{il\ le\ prends\}))$, hence

$\sigma(\text{dom}\tau) \cap \text{Pref}(\varphi_2^{-1}(\{il\ le\})) \subset$

$\sigma(\text{dom}\tau) \cap \text{Pref}(\varphi_2^{-1}(\{il\ le\ prends\})) \neq \varnothing$

and

$F_\tau(il\ le\ prends) = \text{im}f(il\ le\ prends, \tau)$.

Now

Unit PRO il Unit PRO le $\in R(F)$

therefore

<*il PRO:ms3s*> <*le PRO:ms3s*> $\in S(\tau)$

Thus there are $u, v \in A_3^*$ such that:

$u\,v$ = <*il PRO:ms3s*> <*le PRO:ms3s*>

$u \in F_\tau(il\ le)$

$v \in S(\tau)$

but

$u \in F_\tau(il\ le) = \text{im}f(il\ le, \tau) \subset \text{im}f(il\ le\ prends, \tau)$

$= F_\tau(il\ le\ prends)$

and, since $u \neq \varepsilon$, $\alpha(v) \neq$ *Unit PRO il Unit PRO le*, hence

$\alpha(v)$ *Unit V V66 prendre v-t P 2 s prends* $\in R(F)$

therefore

v <*prendre V66:P2s*> $\in S(\tau)$

then

$u\,v$ <*prendre V66:P3s*> $\in$

$F_\tau(il\ le\ prends)(\varphi_2^{-1}(\{\varphi_2(v)\ prends\}) \cap S(\tau))$

so

<*il PRO:ms3s*> <*le PRO:ms3s*> <*prendre V66:P2s*> $\in S(\tau)$:

contradiction.

Similar methods with boundary symbols

The counterexamples above show that none of the methods has a larger expressive power than the other. The differences are related to the recognition of the left and right limits of sentences[3]. It is possible to modify the methods

[3] D. Perrin called my attention to that.

above in order to introduce boundary symbols, so that the resulting methods share exactly the same theoretical expressive power (Laporte, to appear).

6. A practical comparison

From a formal point of view, the methods described above are ways of defining a rational language $L \subset C^*$ (resp. $L \subset A_3^*$) by specifying a finite-state local grammar. Their interest does not come from that, since any rational language can be specified in a such a way (Kleene's theorem). It is rather a practical interest. Local-grammars are hand-made linguistic data: they are useful if they are easy to write and easy to check, i.e. if with a limited size and a good level of readability, they define a superset of *Corr*, a "large" language. Considerations on the ease of designing and checking local grammars are subjective, they depend on the grammatical intuitions of the writer of the grammars. The following is based on our experiments in writing local grammars and testing them with both tools on text corpora.

Designing local grammars

Look at a portion of non-disambiguated tagged text, e.g. in Fig. 5, produced by INTEX: that will probably immediately give you ideas of disambiguation rules. Such spontaneous ideas usually sound like "The determiner *du* cannot be followed by a tensed verb", i.e. some sequence is forbidden, or "If the determiner *du* is followed by a verb, the verb can only be in the present participle", which contains a recognition part and a grammatical constraint. Designing a first version of a local grammar is an easy task, e.g. using INTEX. Depending on the writer of the grammar, the fact that Roche's local grammars contain negative information may be felt as an obstacle; for our own part we did not feel so.

Checking local grammars

Grammatical intuitions may be wrong, e.g. "A noun is not followed by a noun". Our objective of zero silence requires testing carefully the grammars.

Roche's grammars of forbidden sequences can be checked manually, using one's intutition to find counterexamples to them and refine them (this is a faculty that can be trained). Corpora and a tool to analyse and visualise them can be helpful: a forbidden grammatical sequence should not be found in corpora. INTEX can locate in corpora the occurrences of a given grammatical pattern, but even for an actually forbidden sequence, many occurrences (with incorrect tags) may be found. The writer of the grammar can use the occurrences located to find counterexamples. Even with a tagged corpus, this aid cannot easily be turned to an automatic test: the fact that a grammatical sequence is not in a given tagged corpus does not mean it is forbidden. It is to be expected that human intuition can be trained to be more efficient than automatic browsing, for this task as for many others.

The formal definition of the languages specified by Silberztein's local grammars show how the rules of interpretation of these grammars are complex. This formal definition is necessary to know in detail what a grammar will do when applied to texts. However, the way those grammars work is more intuitive than what the formal definition would suggest. Here again, it is possible to train one's intuition to find counterexamples to a grammar and refine it. The complexity of the method comes up e.g. if you have two correct local grammars and want to build a grammar that solves the same ambiguities: in general, their union is not correct and you have to study them in detail.

son	<son.DET:ms> | <son.N:ms>
chien	<chien.A:ms> | <chien.A:ms:fs:mp:fp> | <chien.N:ms>
Pucci	
du	<du.DET:ms>
type	<type.A:ms:fs> | <type.N:ms> | <typer.V:P1s:P3s:S1s:S3s:Y2s>
paresseux	<paresseux.A:ms:mp> | <paresseux.N:ms:mp>
enrobé	<enrobé.N:ms> | <enrober.V:Kms>
dormait	<dormir.V:I3s>

Fig. 5.

Conclusion

The disambiguating tools described above have much in common: in both cases, finite-state local grammars are used to define rational languages, and the grammar is written in such a way that every correct tagged sequence should belong to the rational language. The way to write the grammar depends on the rules which define the rational language as a function of the grammar. Other disambiguating tools could be designed following the same general principle.

We mentioned that such tools can also be used to detect non-lexical errors. Another application is to be foreseen if we follow the same general principle but use tags with syntactic information in addition to grammatical information, including boundary tags. The result of the tagging would then take into account syntactic structure and more lexical ambiguities, and therefore would be a parsing of input text.

References

Berstel, Jean. 1979. *Transductions and Context-Free Languages*, Stuttgart: Teubner, 278 p.

Brill, Eric. 1992. "A Simple Rule-Based Part of Speech Tagger", *3rd Applied ACL*, Trento (Italy), pp. 152-155.

Courtois, Blandine. 1990. "Un système de dictionnaires électroniques pour les mots simples du francais", in *Langue française* no. 87, *Dictionnaires électroniques du francais*, Paris: Larousse, pp. 11-22.

Cutting, Doug, JulianKupiec, Jan Pedersen,Penelope Sibun. 1992. "A practical part-of-speech tagger", *3rd Applied ACL*, Trento (Italy), pp. 133-140.

Federici, Stefano, Vito Pirrelli. 1992. "A Bootstrapping Strategy for Lemmatization: Learning Through Examples", *Papers in Computational Lexicography. COMPLEX 92*, F. Kiefer, G. Kiss, J. Pajzs, eds., Linguistics Institute of the Hungarian Academy of Sciences, Budapest, pp. 123-135.

Garside, Roger, GeoffreyLeech, Geoffrey Sampson. 1987. *The Computational Analysis of English*, Longman.

Hindle, Donald. 1983. "Deterministic parsing of syntactic non-fluencies", *21st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference.*

Hindle, Donald. 1989. "Acquiring disambiguation rules from text", *27th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 118-125.

Koskenniemi, Kimmo. 1990. "Finite-state parsing and disambiguation", *Proceedings of COLING 90*, H. Karlgren, ed., Helsinki University, pp. 229-232.

Milne, Robert. 1986. "Resolving Lexical Ambiguity in a Deterministic Parser", *Computational Linguistics*, vol. 12, no. 1, pp. 1-12.

Paulussen, Hans, Willy Martin. 1992. "DILEMMA-2: a Lemmatizer-Tagger for Medical Abstracts", *3rd Applied ACL*, Trento (Italy), pp. 141-146.

Pelillo, Marcello, Mario Refice. 1991. "Syntactic disambiguation through relaxation processes", *Eurospeech 91*, vol. 2, pp. 757-760.

Revuz, Dominique. 1991. *Dictionnaires et lexiques, methodes et algorithmes*, PhD dissertation, Publication no. 91-44 of LITP, University Paris 7, 105 p.

Rimon, Mori, Jacky Herz. 1991. "The recognition capacity of local syntactic constraints", *5th Conference of the European Chapter of the ACL. Proceedings of the Conference*, Berlin, pp. 155-160.

Roche, Emmanuel. 1992. "Text disambiguation by finite-state automata, an algorithm and experiments on corpora", in *COLING-92, Proceedings of the Conference*, Nantes.

Silberztein, Max. 1989. *Dictionnaires électroniques et reconnaissance lexicale automatique*, PhD dissertation, LADL, University Paris 7, 176 p.

Silberztein, Max. 1990. "Le dictionnaire électronique des mots composés", in *Langue française* no. 87, *Dictionnaires électroniques du francais*, Paris: Larousse, pp. 71-83.

Silberztein, Max. 1993. *Dictionnaires électroniques et analyse automatique de textes. Le système INTEX*, Paris: Masson, 233 p.

Vosse, Theo. 1992. "Detecting and Correcting Morpho-syntactic Errors in Real Texts", *3rd Applied ACL*, Trento (Italy), pp. 111-118.

Un Système d'Interprétation des Verbes Psychologiques du Français

YVETTE YANNICK MATHIEU

Abstract

The aim of this paper is to describe a system for understanding French psychological verbs. We have used a prototype-based formalism and deductive rules for the linguistic and semantic knowledge, with a inheritance mecanism.

1. Présentation des objectifs du système

Nous présentons un système d'interprétation automatique des verbes psychologiques du français. Etant donnée une phrase contenant un de ces verbes, le système permet :
1) de préciser si l'émotion ressentie est agréable ou désagréable et de donner des verbes sémantiquement proches. Ainsi, à partir de :

Luc terrifie Marie

on sait que l'émotion, plutôt désagréable, est ressentie par *Marie*, que la cause ou l'agent en est *Luc*, que l'interprétation est *effrayer beaucoup*, et que les verbes *épouvanter* et *terroriser* peuvent remplacer *terrifier* dans la phrase, en conservant le même sens ;

2) de déduire la sémantique d'un des actants. Considérons la phrase suivante :

(1) *Ce paragraphe heurte les oreilles de Marie*

Le sentiment, plutôt désagréable, ressenti par *Marie* est provoqué par quelque chose qu'elle **entend**. Pour faire cette déduction, le trait sémantique intrinsèque de *paragraphe*, comme **"écrit"**, ne donnerait pas la bonne interprétation, car celle-ci doit être également un **"son"** (*un paragraphe lu*, par exemple). C'est pourquoi dans notre système, c'est à partir du verbe et de son complément, c'est-à-dire *heurter les oreilles* dans l'exemple (1), que l'on peut faire des inférences sur le champ sémantique du sujet, ici un **son** ;

3) d'associer des phrases reliées syntaxiquement. Ainsi, à :

Que Luc ait un tel comportement déçoit Marie

les phrases suivantes sont associées :

Marie est déçue par le fait que Luc ait un tel comportement
Marie éprouve de la déception devant le fait que Luc ait un tel comportement
Marie est déçue de ce que Luc ait un tel comportement
Marie est déçue que Luc ait un tel comportement
Marie est déçue

4) de faire varier l'intensité du verbe. Ainsi, si l'on a :

L'opéra intéresse Marie

une augmentation (progressive et continue) de l'intensité donne :

L'opéra passionne Marie
L'opéra transporte Marie

5) de fournir une interprétation sémantiquement opposée. Ainsi, la phrase :

Les paroles de Luc énervent Ida

a pour contraire :

Les paroles de Luc apaisent Ida

2. Classement sémantique des verbes psychologiques

Nous avons constitué des classes de verbes en fonction du champ sémantique des verbes impliqués. Notre propos n'est pas de rendre compte de la diversité sémantique des verbes psychologiques (ce qui nécessiterait presque une classe par verbe), mais de regrouper les verbes ayant un sens voisin en classes d'équivalence. Ces classes sont d'abord basées sur l'intuition sémantique, puis ont été soumises pour vérification à des locuteurs. S'il y a un large consensus pour regrouper ensemble des verbes tels que *énerver, exaspérer* et *irriter*, d'autres regroupements sont sujets à variations selon les sujets.

Ces classes d'équivalences sont définies par un parangon acceptable qui peut remplacer tous les verbes de la classe. Par exemple la classe de parangon *EFFRAYER* contient les verbes *affoler, alarmer, angoisser, apeurer, effaroucher,* etc. Le verbe choisi comme parangon est d'un niveau de langue courant (et non familier, littéraire ou vieilli), il appartient à une seule classe pour éviter toute ambiguïté et il est "neutre"[1]. Nous avons abouti à une classification où l'on peut distinguer trois grandes catégories de verbes :

-Les verbes qui font ressentir (ou qui causent) un sentiment plutôt désagréable tel que la tristesse, l'ennui, la peur, l'exaspération, etc.,

-Les verbes qui font ressentir (ou qui causent) un sentiment plutôt agréable tel que la joie, l'apaisement, l'émerveillement, la passion, etc.,

---

[1]verbe auquel on peut ajouter le modifieur *beaucoup*

-Les verbes qui font ressentir (ou qui causent) un sentiment ni agréable ni désagréable. Ainsi, nous avons classés 390 verbes qui font partie de la table 4 du lexique-grammaire de Gross 1975, c'est-à-dire qui entrent dans la structure $N_0\ V\ N_1$ où N_1 est un objet humain qui ressent un sentiment déclenché par le sujet N_0. Parmi ces verbes, il y en a 215 "désagréables" regroupés en dix-huit classes, 152 "agréables" regroupés en treize classes et 23 "indifférents" regroupés en deux classes. Ces classes ont une intersection vide. Si un verbe est ambigu, il est considéré comme recouvrant des verbes différents.

A ces classes fondées sur l'intuition sémantique, nous avons associé un ensemble de propriétés syntactico-sémantiques formelles, et un ensemble de règles d'interprétation (thèse en préparation et Mathieu 1995).

3. Représentation des connaissances

Les connaissances à représenter sont de deux natures : des connaissances élémentaires et des connaissances complexes. Par connaissances élémentaires nous désignons, d'une part, les propriétés distributionnelles et aspectuelles des verbes et, d'autre part, les propriétés qui décrivent la possibilité d'associer certaines phrases à la construction de base. Par connaissances complexes nous désignons des connaissances obtenues par la mise en oeuvre d'inférences.

Les connaissances que nous avons à représenter, et les traitements qui leur sont associés, nous ont conduit à adopter un formalisme hybride qui emprunte à plusieurs des représentations utilisés en Intelligence Artificielle. Nous avons choisi une représentation en prototypes (Rosch 1975), organisés de façon hiérarchique à partir d'une racine unique. La racine contient des propriétés communes à toutes les classes sémantiques, par exemple la structure $N_0\ V\ N_1$ qui caractérise les verbes de la table 4 du lexique-grammaire. Chaque classe sémantique est représentée par un prototype auquel sont associés un ensemble de propriétés et une base de connaissances. Ces connaissances sont exprimées par des règles de production du type :

Si	condition(s)
alors	conclusion(s)

Voici par exemple une des règles qui concerne les constructions passives :

Si	la phrase à analyser est de la forme $N_0\ V\ N_1$
et si	le sujet est de la forme *Que P*
alors	la construction N_1 *est Vpp par le fait que P* est acceptable
et	la construction N_1 *est Vpp que P* est acceptable

Que Luc parte si tôt irrite Marie
Marie est irritée par le fait que Luc parte si tôt
Marie est irritée que Luc parte si tôt

Le système comporte 30 règles de production. Des relations d'intensité et d'antonymie entre les verbes sont décrites par des graphes. Un mécanisme d'héritage non monotone permet un partage des propriétés, et des bases de connaissances (Daelemans, De Smedt et Gazdar 1992).

4. Architecture du système

Le système comprend quatre modules principaux :

1) Une interface de communication avec les utilisateurs. Elle est constituée de deux parties :

a) un module de communication destiné à l'analyse d'une phrase. L'accent n'ayant pas été mis sur le dialogue homme-machine, la phrase à analyser peut soit être issue d'un analyseur syntactico-lexical, soit être fournie explicitement par l'utilisateur. Ce module permet :

-de prendre en compte les spécifications des informations fournies au système,

-de rechercher dans la base sémantique le ou les prototype(s) *Pi* du verbe considéré,

-de générer une analyse pour chaque couple (*V* , *Pi*),

-d'afficher les résultats du module de traitement des connaissances pour chaque analyse.

b) un module d'interface pour l'enrichissement de la base de connaissances. Cet enrichissement se fait indépendamment des traitements qui lui sont appliqués.

2) Une base sémantique qui contient le savoir du système, c'est-à-dire, d'une part, l'ensemble des prototypes et de leurs spécialisations, organisés selon une hiérachie arborescente. Chaque prototype et chaque spécialisation contient les propriétés qui lui sont propres et une base de connaissances qui lui est associée. D'autre part, cette base contient des graphes d'intensités sémantiques et d'antonymie, organisés de façon transversale au niveau des prototypes.

3) Un module de gestion d'héritage des propriétés et des connaissances complexes. Etant donnés un verbe et son prototype, ce module construit, pour le temps de l'analyse, d'une part, l'ensemble des propriétés du verbe et d'autre part, sa base de connaissances. L'ensemble des propriétés d'un verbe est formé par propagation, en premier lieu, des propriétés de la racine au prototype du verbe, puis du prototype du verbe au verbe. De façon similaire, la base de connaissances du verbe est construite par propagation de la base de connaissances du prototype *RACINE* puis de celle du prototype du verbe. A chaque niveau d'héritage, il y a prise en compte éventuelle d'une propriété ou d'une règle locale (au prototype ou au verbe) qui masque la propriété ou l'inférence héritée.

4) Un module de traitement des connaissances qui déclenche et gère un ensemble d'opérateurs définis sur la base sémantique. Pour permettre une indépendance des traitements vis-à-vis de la base sémantique, nous avons adopté une structuration du traitement des connaissances sous forme d'opérateurs. Ces opérateurs sont autonomes, chacun est spécialisé pour un type précis de recherche d'information ou de traitement des connaissances. Cette modélisation sous forme d'opérateurs permet un enrichissement permanent de la base sémantique sans avoir à modifier le traitement de ses connaissances. On peut ajouter, modifier ou enlever indifféremment des verbes, des classes sémantiques, des propriétés, des liens d'intensité entre classes ou des règles déductives sans avoir à modifier les opérateurs qui portent sur ces différentes informations.

Conclusion

Notre système utilise une représentation des connaissances qui, grâce à la représentation prototypique et au mécanisme de l'héritage, vise à l'économie de la description. Le formalisme adopté, par son aspect déclaratif, permet une grande lisibilité et un enrichissement progressif des connaissances. Nous avons appliqué notre système aux verbes psychologiques, mais sa conception en fait un outil très général de modélisation pour les données linguistiques qui ont des caractéristiques proches de celles que nous avons traitées. La faculté de créer de nouveaux opérateurs sur cette structure de connaissances constitue l'ébauche d'un langage déclaratif destiné au traitement de la sémantique des verbes.

Références

Daelemans, Walter; Koenraad De Smedt; Gerald Gadzar. 1992. Inheritance in Natural Language Processing. *Computational Linguistics* 18(2), Rochester: USA.

Gross, Maurice. 1975. *Méthodes en syntaxe*. Paris: Hermann.

Mathieu, Yvette Yannick. 1995, à paraître. Verbes psychologiques et interprétation sémantique. *Langue française*, Paris: Larousse.

Rosch, Eleanor. 1975. Cognitive Representation of Semantic Categories. *Journal of Experimental Psychology* 104(3).

Syntactic Analysis by Local Grammars Automata: an Efficient Algorithm

MEHRYAR MOHRI

Abstract

Texts contain many ambiguities. The description of the constraints restricting the word order for specific context provides helpful grammars for solving this problem. Indeed, these grammars allow to easily eliminate many of the ambiguities without even using complex general syntactic rules involving a lexicon-grammar. Local grammars can be represented in a very natural way by finite state automata. This paper describes and illustrates an efficient algorithm which allows to apply local grammars to the automaton representing the text.

1. Introduction

One of the main incentives of syntactic analysis is to eliminate irrelevant ambiguities of a text. Local grammars constitute useful descriptions which help to remove some of these ambiguities. They consist of the description of local constraints, namely restrictions on the surrounding sequences of a given set of words. Combinations of French pre-verbal particles (see M. Gross 1989), in some extent agreement rules, other constraints independent of lexicon-grammar's entries, and many rules useful for error correction in texts provide typical examples of local grammars[2].

As shown further, local grammars can be represented in a very convenient way by finite state automata. The corresponding automata describe sets of locally unacceptable sequences[3] that a correct text should not contain. Once tagged, a text can in fact be itself represented by an automaton. Each of its each path then constitutes an ambiguity. Thus, checking its correctness consists in removing the paths containing any of the forbidden sequences of the local grammar's automaton.

This requires searching in the automaton of the text for all occurrences of sequences of the local grammar's automaton. Such an operation can be considered as a generalization of the classical string matching problem which consists in finding all occurrences of a word in a text: here, we need to find a set of sequences in a set of texts. Several efficient algorithms have been proposed for solving the problem of locating occurrences of a finite set of sequences in a single text (A. V. Aho and M. J. Corasick 1975, B. Commentz-Walter 1979), and the application of local grammars to texts has already been described by E. Roche (1992).

---

[1] Laboratoire d'Automatique Documentaire et Linguistique and Institut Gaspard Monge.

[2] See M. Rimon and J. Herz (1991), and, F. C.N. Pereira and R. N. Wright (1991) for other related use of automata in syntactic analysis, and D. Maurel (1989) for a description of time expressions constraints in French by local grammars.

[3] Local grammars can be represented in an equivalent way by the set of obligatory sequences.

Here, we shall present a more efficient algorithm with a better time complexity which uses the notion of *failure function* or *default function* brought in by A. V. Aho and M. J. Corasick (1975) and extends it to the representation of automata. Analogous extensions have already been operated by M. Crochemore (1986). They show failure functions to be a helpful notion in the representation of automata.

In the following, we first illustrate the application of local grammars by considering several examples, then give a complete description of our algorithm and indicate corresponding experimental results.

2. Application of local grammars

Simple local rules can be easily represented by finite state automata. Consider, for instance, the word *this* in English. It is ambiguous for it can be a determiner, a demonstrative adjective, as in the following sentence:

This program works well,

a demonstrative pronoun as in:

This does not change his opinion,

or an adverb:

He is not this tall.

However, *this* imposes constraints to the choice of words it precedes. Simple observations lead to the following rules:

i) when *this* is a determiner it cannot be followed by a verb unless the verb is a past or present participle;

ii) the adverb *this* cannot be followed by a noun nor by a verb.

Rule *i*) can be illustrated by the following sentences:

* *This sing is pretty*
This falling rock is dangerous
He hates this inherited impatience of Lea.

When *this* is an adverb, it can be followed by an adjective or by some other adverbs like *much* as in:

He is not this much cleverer than her[4].

Thus, the second rule indicates only some of the forbidden sequences in this case.

---

[4] Replacing *this* by *that* makes this sentence stylistically better.

Notice that the above rules are expressed in a negative way. Thus, it is quite natural to represent them by an automaton storing the set of unacceptable combinations. Figure 1 illustrates this automaton[5].

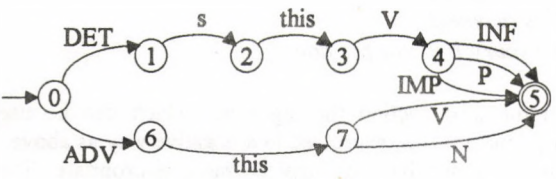


Figure 1. Local grammar of the form *this*.

It can be used to eliminate some of the ambiguities encountered in a text. Consider the sequence *this limit* which can be found in various contexts. *limit* is also an ambiguous form as it can be a verb conjugated at present at any person except the third of singular or an infinitive or an imperative form, or a singular noun. Thus, a simple dictionary look up allows to represent this sequence by the following automaton. Each path from the first state 0 to the final state 6 constitutes a possible analysis. However, the local grammar above helps to climinate some of these ambiguities. Each path containing a sequence of the corresponding automaton can be removed.

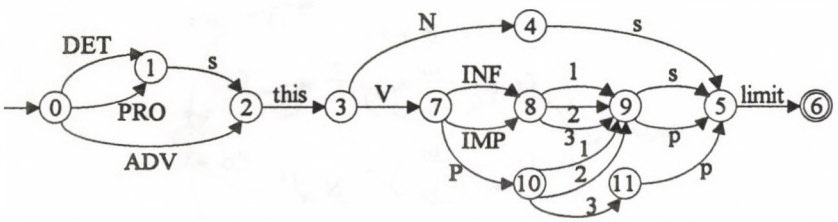


Figure 2. Automaton of the text *this limit*.

Thus, the application of the local grammar must lead to the following automaton.

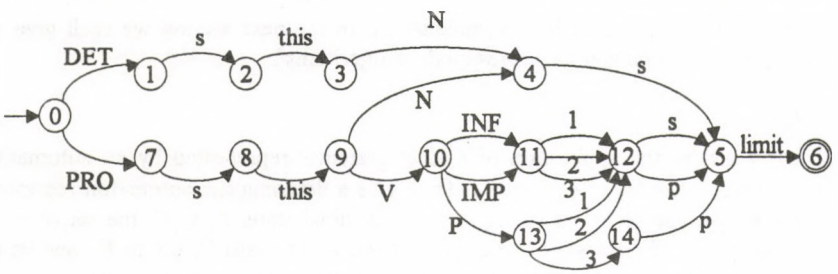


Figure 3. Automaton obtained after application of the local grammar.

[5] Here, labels *p* and *s* stand for plural and singular, *ADV* for adverb, *DET* for determiner, *N* for noun, *PRO* for pronoun, *IMP* for imperative, *INF* for infinitive, *P* for present, *V* for verb and *1*, *2* and *3* indicate the first, second and third person. In order to simplify this presentation, we do not take into account here the subjunctive.

Notice that many of the remaining paths can be part of acceptable sentences. The following sentences illustrate some of these sequences:

He made this limit the disaster
Let this limit his rudeness
After this, limit yourself to one per day.

We shall describe in the next section the algorithm which can be used to perform the application of a local grammar when expressed in a negative way as above. However, in some cases, expressing rules in a positive way may be more appropriate. This can occur when acceptable sequences are less numerous or easier to describe than forbidden ones.

Agreement rules are often more easily expressed this way. Figure 4 gives a sample of agreement rules in French concerning articles *un* and *le* represented by an automaton. This automaton contains a set of obligatory sequences. It should be read in the following way: if *le* or *un* is a determiner masculine singular[6] followed by a noun, then this noun must also be masculine singular (paths '0 1 2 3 4 5'). Notice that this automaton gives no information about the constraints concerning a case where one of these determiner is followed by an adjective instead of a noun, and that it imposes no restriction on a sequence which does not contain these articles.

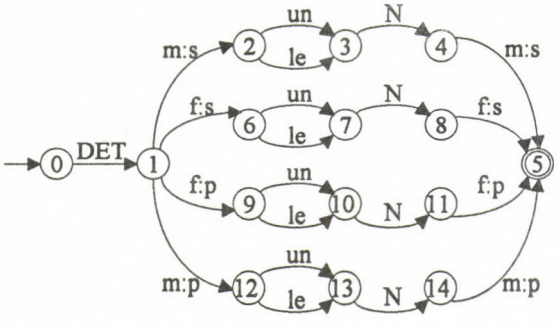


Figure 4. Agreement automaton for French articles *un* and *le*.

Such automata can also be used for disambiguation. In the next section we shall give more details about their definition and the corresponding algorithms[7].

3. Algorithm

We shall first consider the application of a local grammar represented by an automaton of forbidden sequences. Let $G_1 = (V_1, i_1, F_1, A^*, \delta_1)$ be a deterministic automaton representing the text, where V_1 is the set of its states, $i_1 \in V_1$ its initial state, $F_1 \subseteq V_1$ the set of its final states, A its alphabet, and δ the state transition function which maps $V_1 \times A$ to V_1, and let $G_2 =$

---

[6] In the automata presented here we are only concerned with the canonical form of each word and with its morphological characteristics. Therefore, the feminine singular article *la* whose canonical form is *le* for instance is denoted by the sequence *DET f:s le*.

[7] Local constraints can also be represented by transducers (see M. Silberztein 1989). The minimization algorithm for transducers can help to limit the size of such transducers (see M. Mohri 1994). Here, we are only concerned with efficient algorithms involving automata.

$(V_2, i_2, F_2, A^*, \delta_2)$ be the automaton representing the local grammar with analogous notations. In order to simplify the following algorithms we shall assume that G_2 is acyclic[8]. We denote by $L(G_1)$ (resp. $L(G_2)$) the language recognized by G_1 (resp. G_2). Thus, $A^*L(G_2)A^*$ constitutes the set of all sentences which contain an unacceptable sequence of $L(G_2)$. The application of the local grammar G_2 to G_1 should then lead to the regular language $L(G_1)\backslash A^*L(G_2)A^*$, namely the set of sentences of $L(G_1)$ which have no factor in $L(G_2)$. Here, we need to define an automaton $G = (V, i, F, A^*, \delta)$ recognizing this language[9].

In order to do so, we shall first indicate how to compute from G_2 a deterministic automaton representing the language $A^*L(G_2)$, namely the set of all sentences which end in $L(G_2)$.

3.1. Construction of a deterministic automaton recognizing $A^*L(G_2)$ from G_2

In general, constructing such an automaton is not a trite operation. It is easy to design a non-deterministic automaton recognizing $A^*L(G_2)$. Indeed, a simple loop labelled by all elements of the alphabet A added at the initial state of G_2 is enough to transform it into an automaton recognizing $A^*L(G_2)$. The same can be done at the final states to obtain a non-deterministic automaton recognizing $A^*L(G_2)A^*$. However the use of this automaton makes the whole operation of application of the local grammar inefficient. Notice that the size of the alphabet A is superior to the one of a dictionary of simple words of the language. Moreover, in some cases as in error correction applications the whole list of the elements of A may not be available. Thus, a simple determinization of this automaton can be time consuming and even impossible in some case.

In order to construct a deterministic automaton recognizing $A^*L(G_2)$, we shall use the notion of failure function and gradually modify the automaton G_2. Consider a sequence $w \in A^*$. In order to know whether w is in $A^*L(G_2)$ we can try to read it using the automaton G_2. As long as there is a transition corresponding to the read word in G_2 we use this transition to step to the following state. This allows us to read a prefix x of w in G_2. If the sequence w is entirely read this way ($x = w$) and the reached state is a final state, then w is in $L(G_2)$ and a fortiori in $A^*L(G_2)$. If not, then w may have a suffix v in $A^*L(G_2)$ (see figure 5). As shown by the figure below, x has then a suffix x' which is a prefix of v. In order to check the existence of a sequence like v, we need to start at a position as much at left as possible. In other words, we need to check whether v is in $L(G_2)$ when v is such that: x' is the longest proper suffix of x which is also a prefix of a sequence of $L(G_2)$.

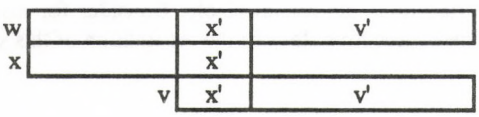


Figure 5. The definition of the failure function.

---

[8] The algorithms presented here can be easily modified in order to handle the case of non acyclic automata. However, local grammars are generally represented by acyclic automata. Besides, G_1, the text automaton, is of course also acyclic although we will not use this condition in the following.

[9] Notice the similarity of this problem with the one of string matching which consists of finding an occurrence of a word x in a text t and which can be expressed by: $t \in A^*xA^*$?

This leads to the definition of a function which associates to each state u of G_2 the longest proper suffix of the paths reaching u which is in G_2. However, this function can only be well-defined if all paths reaching u have the same longest proper suffix x' in G_2. Thus, in case two paths of G_2 leading to u have different longest proper suffix x' in G_2, we need to duplicate u in two states corresponding to each of these paths. So, we can define a *failure function* s, which associates with each state u the state corresponding to x', namely $\delta_2(i_2, x')$. This function is to be consulted whenever the desired transition does not exist at a given state u (A. V. Aho and M. J. Corasick 1975). Thanks to the use of a failure function it is possible to represent the desired automaton even if the alphabet A is infinite or undefined.

The following figures give an example of an automaton G_2 and its associated automaton G_3 which represents $A^*L(G_2)$.

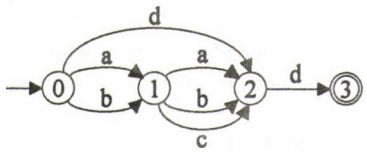


Figure 6. Local grammar G_2.

State numbers on the graph of figure 7 are followed by a slash and the value of the failure function at that state. For example, we have $s[4] = 1$, as the longest proper suffix of the sequence *aa* which is recognized by G_2 is *a*, the state corresponding to *a* is 1, and considering other combinations *ab*, *ba*, *bb* leads to the same result.

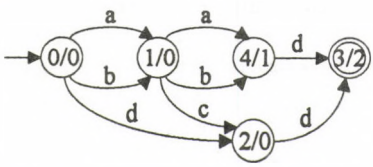


Figure 7. G_3, a deterministic automaton for $A^*L(G_2)$.

Notice that the construction of this automaton has required the duplication of the state 2 of G_2, as *bb* and *bc* do not have the same longest proper suffix in G_2 (resp. *b*, and the empty word ε). The recognition process by such an automaton is quite easy. One just needs to use default transitions whenever usual transitions are not available. Consider for instance the sequence *aacd* which is in $A^*L(G_2)$. The consecutive steps of the recognition of this sequence are:

0 *aacd*
1 *acd*
4 *cd*
1 *cd* (failure transition)
2 *d*
3 *ε*.

As 3 is a final state, the sequence *aacd* is correctly recognized by $G_3=(V_3, i_3, F_3, A^*, \delta_3)$. It can be easily showed that the recognition process of a sequence w can still be done in $O(|w|)$ when using a representation by default functions.

The failure function s can be computed in an easy way. Indeed, it can be proved that for any state u and any element a of the alphabet A, $s[\delta(u, a)]$ is the first $\delta(s^k[u], a)$, $(k \geq 1)$, such that $\delta(s^k[u], a)$ is defined, or the initial state if none of these is defined. This gives a recursive algorithm for calculating this function. Notice that the definition of the failure function involves proper suffixes, hence for any state u, the level of the state $s[u]$ is lower than the one of u. In order to check whether a transition $\delta(s^k[u], a)$ is defined we then need to have defined and computed s for all states v with lower levels than u. This restricts the ordering in which the states should be considered in the algorithm. A breadth-first search (A. V. Aho *et al.* 1974, B. Sedgewick 1988, T. H. Cormen *et al.* 1990) of the automaton meets the corresponding condition. It suggests the use of a first-in first-out queue Q for managing the set of states to visit at each step. Figure 8 gives a pseudocode for an algorithm computing a deterministic automaton recognizing $A^*L(G_2)$ from G_2.

```
1   for each u ∈ V(G2)
2     do  s[u] ← UNDEFINED
3   Q ← {i}
4   s[i] ← {i}
5   while Q ≠ ∅
6     do  u ← head[Q]
7         for each t ∈ Trans[u]
8               do  v ← s[u]
9                   while v ≠ i and δ(v, t.l) = UNDEFINED
10                      do v ← s[v]
11                  if u ≠ i and δ(v, t.l) ≠ UNDEFINED
12                      then  v ← δ(v, t.l)
13                  if s[t.v] = UNDEFINED
14                      then  s[t.v] ← v
15                            ENQUEUE(Q, t.v)
16                            LIST-INSERT(list[t.v], t.v)
17                  else if there exists w ∈ list[t.v] such that s[w] = v
18                      then  t.v ← w
19                      else  w ← COPY-STATE(t.v)  ◊copy of t.v with same transitions
20                            s[w] ← v
21                            LIST-INSERT(list[t.v],w)
22                            t.v ← w
23                            ENQUEUE(Q, w)
24        DEQUEUE(Q)
```

Figure 8. Algorithm for the construction from G_2 of a deterministic automaton for $A^*L(G_2)$.

We here denote by $Trans[u]$ the set of transitions leaving a state $u \in V$, and for each t in $Trans[u]$ and $u \in V$, by $t.v$ the vertex reached by t and $t.l$ its label. We also use a special constant UNDEFINED different from all states of G_2. The algorithm directly modifies the automaton G_2 into one representing $A^*L(G_2)$, by duplicating states whenever it is necessary (function COPY-STATE), and by computing the failure function s for all states. In order to limit the duplication of states, the list of copied states of a state u is stored at each step in $list[u]$ and a new state is created only if no other equivalent state with the same default state exists.

Notice that the loop of lines 5-23 iterates as long as there remains a state u for which all leaving transitions have not yet been examined. Each state u is enqueued exactly once in Q. Hence, the total number of iterations of the loop 5-23 is equal to the number of states of the resulting automaton G_3. The loop of lines 7-22 is performed once for each transition leaving u. Thus, if we denote by $E(G_3)$ the set of the transitions of G_3, in the whole this loop is iterated $|E(G_3)|$ times. If the test made at line 17 and the insertion of line 16 are efficiently implemented by using hashing method, each iteration can be assumed to be done in constant time. The total running time of the algorithm including the initialization (lines 1-2) is then $O(|V(G_3)| + |E(G_3)|)$, thus linear in the number of states and transitions of the resulting automaton.

It is also worthwhile to point out that if the automaton G_2 is minimal[10], then the resulting automaton G_3 is also the minimal deterministic automaton representing $A^*L(G_2)$. Indeed, if two states u and u' were equivalent in G_3, then by definition[11] they would be copies of a same state v of G_2. As states of G_2 are duplicated only if necessary, then u and u' bear different failure function's values. Therefore, different sequences can be read from u and u' to a final state of G_3. This contradicts the equivalence of these states.

There are some particular cases in which the size of the obtained automaton G_3 is exponential. The minimal automaton associated to $a(a + b)^n$ for instance has $(n + 2)$ states whereas it is easy to show that the minimal automaton of $(a + b)^*a(a + b)^n$ has 2^{n+1} states. However, such blow up cases are generally not encountered in Natural Language problems, and if they could occur then the result of the application of the corresponding local grammar could also have an exponential size.

In the following section, we shall indicate how to use the obtained automaton recognizing $A^*L(G_2)$ so as to apply the local grammar.

3.2. Application of the local grammar and experimental results

Once the automaton G_3 representing $A^*L(G_2)$ is provided, the application of the local grammar G_2 becomes considerably easier. Given an automaton G_1 representing a text, one can directly construct a deterministic automaton corresponding to the language $L(G_1)\backslash A^*L(G_2)A^*$, by using G_1 and G_3. Indeed, we can simultaneously read these two automata, store at each step the two states reached in each of them and keep those transitions of G_1 which do not lead to a final state of G_3. This can be illustrated by the following figures. Figure 9 gives an example of a text automaton, and figure 10 the automaton G_4 obtained by using G_3 (figure 7). Each state of G_4 bears a pair of numbers indicating the states reached respectively in G_1 and G_3.

The initial state of G_4 corresponds to the pair (0,0) of initial states of G_1 and G_3. Transitions a and b, for instance both lead from this state to (1,1) as reading these transitions lead to state 1 in G_1 and also 1 in G_3.

---

[10] Notice that minimal automata representing local grammars of unacceptable sequences have only one final state, as there is no need adding an unacceptable sequence to such local grammars when a prefix of it is already part of forbidden sequences.

[11] Notice that if we do not take into account default transitions, then G_3 represents the same language as G_2.

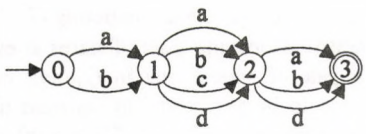


Figure 9. Text automaton G_1.

The transition labelled by a from (1,1) leads to (2,4) as $\delta_1(1, a)=2$ and $\delta_3(1, a)=4$. Transitions by d from (2,4) and (2,2) are not kept (represented by dotted line) as they lead to the final state 3 of G_3.

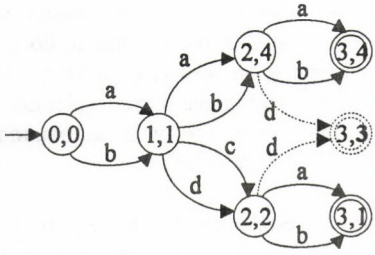


Figure 10. Automaton G_4 obtained from G_1 by application of the local grammar G_2.

This construction is similar to the one used to obtain the intersection of two automata.

```
LOCAL-GRAMMAR(G1, G3, G4)
1   F4 ← ∅
2   {i4} ← (i1, i3)
3   Q ← {i4}
4   while Q ≠ ∅
5      do u4=(u1, u3) ← head[Q]
6         for each t ∈ Trans[u1]   ◊ transitions considered in G1
7            do v1 ← δ1(u1, t.l)
8               v3 ← u3
9               while v3 ≠ i3 and δ3(v3, t.l) = UNDEFINED
10                 do   v3 ← s[v3]
11              if δ3(v3, t.l) ≠ UNDEFINED
12                 then v3 ← δ3(v3, t.l)
13              if v3 ∉ F3
14                 then v4 ← (v1, v3)
15                      if v4 is a new state
16                         then ENQUEUE(Q, v4)
17                              if v1 ∈ F1
18                                 then F4 ← F4 ∪ {v4}
19                      δ4(u4, t.l) ← v4
20        DEQUEUE(Q)
```

Figure 11. Algorithm for the application of a local grammar.

The simple pseudocode above gives the algorithm computing G_4. This algorithm is efficient as it does not require to inspect transitions leaving a set of states at each step of its execution but only those corresponding to a pair of states, one in G_1 and one in G_3. In case the test performed at line 15 is considered to be performed in constant time[12], the algorithm can be showed to be quadratic, more precisely in $O(|V(G_3)|.(|V(G_1)| + |E(G_1)|))$.

We have implemented and experimented this algorithm and the one presented in the previous section. We have tested these algorithms by considering a set of 1.600 sequences of length 20 or more. The corresponding minimal automaton had about 18.000 states. We then defined a simple automaton of about 290 states so as to simulate a local grammar[13].

The first algorithm applied to this automaton led to an automaton of about 340 states. We have checked the fact that the number of states of the automaton do not grow exponentially after application of this algorithm by carrying on several experiments with automata reaching the size of about 2500 states. In our experiments, the number of states of the resulting automaton never exceeded one and a half the one of the initial automaton. The time spent for the execution of this algorithm never reached the second[14].

Notice that once the operation corresponding to this algorithm has been performed the resulting automaton can be used for disambiguating any text. Hence, the time spent for the construction of this automaton can be considered as pre-processing of the grammar and done once for all. The two algorithms described can be combined into a single one such that only necessary states of the automaton G_2 be considered and that only corresponding failure function values be evaluated. The previous remark, however, reduces the interest of such an algorithm.

The application of the local grammar to the text following the algorithm above results in an automaton which had about 17300 states. More precisely, this figure corresponds to the automaton obtained after minimization. The whole process of application of the local grammar, including this minimization, took about 10'. This suggests that the algorithm presented above is very efficient for the sizes we considered.

3.3. Local constraints described in a positive way

We have already indicated the possibility of representing the set of local obligatory sequences by automata. Here we shall give more details about their exploitation.

Positive rules are often expressed in the following way: if an expression contains the sequence X, then it must be followed by the word y, or the last word of X must have the property z. This appears clearly in agreement rules as in the example indicated previously for French articles followed by a noun. Hence, the paths of an automaton corresponding to positive rules do not

---

[12] This is roughly the case if we use an efficient hashing method for the implementation.

[13] The number of states of the local grammars we had at our disposal did not exceed fifty. This simulation aimed at anticipating the fast growth of creation of definition of new local grammars. Their union could very soon reach several hundreds of states.

[14] Our program was written in C and implemented on a Next Cube, processor 68040 33Mhz with 32 Mb of RAM, and on a IBM PS/2 i486 50 Mhz with 16 Mb of RAM.

necessarily constitute all possible paths in a given context. They can only be used in the following way: if the beginning X of one of these paths is encountered in a text and if Xy leads to a final state, then the corresponding part of the text must also be followed by the label y or another label y' such that Xy' be the beginning of a path in the local grammar automaton. Thus, final states play an important role in positive local grammar automata.

The automaton below represents a part of the agreements on gender and number of the French article *un* with the adjectives or nouns following it. The notation '?' is here used to represent any possible canonical form in this context.

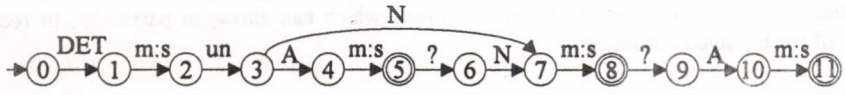


Figure 12. Local grammar for part of agreements of article *un*.

It can be read in the following way: if *un* is a determiner masculine singular followed by an adjective, then this adjective is also masculine singular, etc. This allows a natural positive representation of some local constraints. Indeed, here, one only needs to describe as much as possible the context of a given sequence, and then impose corresponding constraints by using final states.

```
POSITIVE-LOCAL-GRAMMAR(G1, G3, G4)
1   F4 ← ∅
2   {i4} ← (i1, i3)
3   Q ← {i4}
4   while Q ≠ ∅
5      do u4=(u1, u3) ← head[Q]
6         for each t ∈ Trans[u1]   ◊ transitions considered in G1
7            do if there exists t' ∈ Trans[u3] such that t'.l = t.l or for any t' ∈ Trans[u3] t'.v
                  ∉ F3
8                  then v1 ← δ1(u1, t.l)
9                       v3 ← u3
10                      while v3 ≠ i3 and δ3(v3, t.l) = UNDEFINED
11                            do v3 ← s[v3]
12                      if δ3(v3, t.l) ≠ UNDEFINED
13                            then v3 ← δ3(v3, t.l)
14                      v4 ← (v1, v3)
15                      if v4 is a new state
16                            then ENQUEUE(Q, v4)
17                                 if v1 ∈ F1
18                                    then F4 ← F4 ∪ {v4}
19                      δ4(u4, t.l) ← v4
20      DEQUEUE(Q)
```

Figure 11. Algorithm for the application of a positive local grammar.

The application of such automata is close to the one described previously. Here too, we shall use the first algorithm presented above in order to construct an automaton G_3 recognizing $A^*L(G_2)$ from G_2. Only the application of G_3 slightly differs from the one indicated above. Instead of keeping those transitions of G_1 which do not lead to a final state of G_3, here we shall reject only those which do not exist in G_3 whereas this graph contains another transition leading to a final state. The corresponding algorithm can be obtained easily from the one indicated above. Figure 11 describes this algorithm.

This algorithm has obviously the same complexity as the one presented above. Therefore, the use of negative or positive rules in the representation of local grammars have no algorithmic effect on their application, and the choice of the appropriate representation should be mainly motivated by practical or heuristical considerations which can allow, in particular, to reduce the size of the involved automata.

4. Conclusion

The number of local grammars allowing to represent more conveniently contextual constraints keeps increasing. Hence, so does the size of the union of all the corresponding automata. The algorithms described here should allow to apply efficiently such automata even with large sizes in order to reduce the number of ambiguities of texts. They also make it more natural to use automata to impose constraints on factors of a text.

Many other operations related to syntactic analysis by automata such as intersections of the form $(A^*L(G)A^* \cap G')$ involve the computation for a given automaton G of a deterministic one representing $A^*L(G)$. The presented algorithms can also improve the efficiency of these operations. They can also be used in other applications such as pattern matching when the provided data is not a list of words to search for in a text, but an automaton representing these words.

5. References

Aho, Alfred V.; John E. Hopcroft; Jeffrey D. Ullman. 1974. *The design and analysis of computer algorithms*. Reading, Mass.: Addison Wesley.

Aho, Alfred V.; Margaret J. Corasick. 1975. Efficient String Matching: An Aid to Bibliographic Search, *Communication of the Association for Computing Machinery 18 (6)*: 333-340.

Aho, Alfred V.; Ravi Sethi; Jeffrey D. Ullman. 1986. *Compilers, principles, techniques, and tools*. Reading, Mass.: Addison Wesley.

Commentz-Walter, B. 1979. A string matching algorithm fast on average. *Automata, Languages and Programming, Lecture Notes in Computer Science 6*, Springer-Verlag, Berlin: 118-132.

Cormen, Thomas H.; Charles E. Leiserson; Ronald L. Rivest. 1990. *Introduction to Algorithms*. 2nd edition, Cambridge, Mass.: The MIT Press, New York: MacGraw-Hill Book Company.

Crochemore, Maxime. 1986. Transductions and repetitions. *Theoretical Computer Science 45*: 63-86.

Eilenberg, S. 1974. *Automata, Languages, and Machines*. Volume A, New York: Academic Press.

Gross, Maurice. 1989. The Use of Finite Automata in the Lexical Representation of Natural Language. *Electronic Dictionaries and Automata in Computational Linguistics, Lecture Notes in Computer Science 377*, Springer-Verlag, Berlin: 34-50.

Maurel, Denis. 1989. *Reconnaissance de séquences de mots par automates, Adverbes de date du Français*. Université Paris 7, PhD Thesis, Paris: LADL.

Mohri, Mehryar. 1993. *Analyse et représentations par automates de structures syntaxiques composées*. Université Paris 7, PhD Thesis, Paris: LADL.

Mohri, Mehryar. 1994. Compact Representations by Finite-State transducers. *32nd Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, Las Cruces, New Mexico.

Pereira, Fernando C.N.; Rebecca N. Wright. 1991. Finite State Approximation of Phrase Structure Grammars. *29th Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, Berkeley, California.

Rimon, Mori; Jacky Herz. 1991. The Recognition Capacity of Local Syntactic Constraints. *Fifth Conference of European Chapter of Association for Computational Linguistics, Proceedings of the Conference*, Berlin.

Roche, Emmanuel. 1992. Text Disambiguation by Finite State Automata, an Algorithm and Experiments on Corpora, *COLING-92, Proceedings of the Conference*, Nantes.

Sedgewick, Bob. 1988. *Algorithms*. 2nd edition, Reading, Mass.: Addison Wesley.

Silberztein, Max 1989. *Dictionnaire électronique et reconnaissance lexicale automatique*. Université Paris 7, PhD Thesis, Paris: LADL.

Représentation de la Combinatoire des Variantes Consonantiques et Vocaliques et de la Combinatoire des Suffixes de Conjugaison des Adjectifs en Coréen

NAM JEE-SUN

Abstract

We present here two types of combinatorial systems for korean adjectives : the system of consonantal and vocalic variants of stems of adjectives, and the system of conjugation suffixes. In the first system, the series contain variants that scarcely differ from each other syntactically or semantically. These series have a great productivity. This phenomenon is particularly productive for adjectives, but it is also observed for verbs and adverbs. The combinatorial system of adjectival or verbal suffixes, which is exceedingly complex, easily gives rise to hundreds of inflected forms. The regrouping and systematic description of these consonantal and vocalic variants is therefore undeniably important. Each such combination, in the form of a local grammar, allows us to elaborate a reliable and significant data set for the constitution of an electronic dictionary.

0. Présentation des données

0.1. Variantes consonantiques et vocaliques

Nous présentons ici quelques séries de variantes consonantiques et vocaliques d'adjectifs en coréen. Il s'agit d'adjectifs qui décrivent la couleur, la forme, l'effet sonore ou le toucher. Les variantes dans chaque série ne diffèrent guère du point de vue sémantique ou syntaxique. On observe, par exemple, 18 expressions qui correspondent à la forme [long] :

(1)

기나길다	*kinakilta*	길쭉길쭉하다	*kilc'ukkilc'ukhata*
기다랗다	*kitalahta*	길쭉스레하다	*kilc'uksîlehata*
기다마하다	*kitamahata*	길쭉스름하다	*kilc'uksîlîmhata*
기르스레하다	*kilîsilehata*	길쭉하다	*kilc'ukhata*
기르스름하다	*kilîsilîmhata*	길쯔막하다	*kilc'îmakhata*
길고길다	*kilkokilta*	길쯤길쯤하다	*kilc'îmkilc'îmhata*
길다	*kilta*	길쯤하다	*kilc'îmhata*
길다랗다	*kiltalahta*	길찍길찍하다	*kilc'ikkilc'ikhata*
길디길다	*kiltikilta*	길찍하다	*kilc'ikhata*

Certaines de ces expressions sont presque identiques, à certaines nuances phoniques près. Soit:

(2)

길쭉길쭉하다	*kilc'ukkilc'ukhata*
길쯤길쯤하다	*kilc'îmkilc'îmhata*
길찍길찍하다	*kilc'ikkilc'ikhata*

Dans le paradigme (2), il n'existe pratiquement pas de différence de sens ou de comportement syntaxique.

D'autres des expressions (1) n'ont pas tout à fait les mêmes traits sémantiques et/ou les mêmes propriétés syntaxiques. Par exemple, on observe les différences de distribution :

(3) (이 밤+민우가 인아를 뒤쫓는 장면)-이 너무 (길+길고길+*기다랗+*길쭉하)-다
(i **ce** *pam* **nuit** + *Minu - ka* **nmtf** *Ina - lîl* **Acc** *twic'och* **poursuivre** *- nîn* **Sd** *cangmyôn* **scène)** *- i* **nmtf** *nômu* **trop** *(kil + kilkokil + *kitalah + *kilc'ukha) - ta* **St**
((Cette nuit + La scène que[où] Minu poursuit Ina) est trop longue)

(4) (책상+막대기)-가 (길+길고길+기다랗+길쭉하)-다
(chăksang **table** + *maktăki* **bâton)** *- ka* **nmtf** *(kil + kilkokil + kitalah + kilc'ukha) - ta* **St**
((La table + Le bâton) est long(ue))

Les adjectifs comme *kitalahta* ou *kilc'ukhata* n'acceptent ni un sujet-complétive ni un sujet abstrait : ils exigent un sujet non-humain concret comme dans (4).

Ces variantes ont une productivité abondante et assez systématique. Par exemple, il existe une centaine d'adjectifs dont le trait sémantique est [noir] (nous en présentons la liste dans l'annexe 1). On y retrouve certains des suffixes observés dans la série (1). Mais, le problème est que l'on ne peut pas prévoir systématiquement le nombre des variantes et leurs formes pour chaque série : la production fait intervenir les particularités morphologiques de chaque élément lexical. Ainsi, les adjectifs de couleur [jaune] et [blanc] sont environ 20, alors que [rouge] prend environ soixante-dix formes et [bleu] une quarantaine. La description ne peut donc être que lexicale.

0.2. Séries des suffixes de conjugaison

Notons que les adjectifs coréens, comme les verbes, prennent directement des suffixes de conjugaison (i.e. sans copule de type *être*). La phrase simple en coréen a donc une structure de type :

N0-nmtf W (V + Adj)-St (N0 (V + être Adj) W)

(où ***nmtf*** indique la postposition du nominatif, ***St*** est un suffixe terminal du mode déclaratif et ***W*** indique une séquence éventuelle de compléments).

En fait, la combinatoire des suffixes pour un élément lexical est très variée : plusieurs séries de paradigmes interviennent. Les combinaisons entre suffixes pour constituer des formes fléchies pourraient se résumer de la manière suivante :

A	*B*	*C*	*D*	*E*	*F*	*G*
Racine	***Suffixe d'honorification du sujet***	***Suffixes de temps du passé ou du futur***	***Suffixes d'aspect ou de modalité***	***Suffixe d'honorification de l'interlocuteur***	***Suffixes terminaux ou déterminatifs ou complémenteurs ou conjonctifs***	***Suffixes de modalité***

Figure 1

L'emploi des éléments des paradigmes obéit à des contraintes qui tiennent compte des conditions morpho-syntaxiques d'emploi d'un élément lexical donné. Par ailleurs, chaque paradigme comportant un certain nombre de séries (par exemple, F en contient une soixantaine à cause d'une cinquantaine de suffixes conjonctifs (comme les suffixes de subordination de *condition*, de *concession* ou de *cause*)), le choix d'un élément dans une de ces séries n'est pas indépendant des choix faits dans les séries précédentes et subséquentes : de tels interférences entre paradigmes sont extrêmement complexes. Par exemple, pour un adjectif comme *pulhänghata* (malheureux), on observe au moins 1800 formes fléchies (on présente dans l'annexe 2 un échantillon de la liste de ces formes).

1. Exemple

1.1. Combinatoire des variantes de l'expression [*courbe*]

Voici la liste des adjectifs qui décrivent la forme [courbe] :

(1)

고부랑하다	*kopulang-hata*
고부랑고부랑하다	*kopulangkopulang-hata*
고부스름하다	*kopusilïm-hata*
고부스레하다	*kopusile-hata*
고부슴하다	*kopusïm-hata*
고부장하다	*kopucang-hata*
고부장고부장하다	*kopucangkopucang-hata*

고불고불하다	*kopulkopul-hata*
고불탕하다	*kopulthang-hata*
고불탕고불탕하다	*kopulthangkopulthang-hata*
고불퉁하다	*kopulthong-hata*
고붓하다	*kopus-hata*
고붓고붓하다	*kopuskopus-hata*
꼬부랑하다	*k'opulang-hata*
꼬부랑꼬부랑하다	*k'opulangk'opulang-hata*
꼬부스름하다	*k'opusîlîm-hata*
꼬부스레하다	*k'opusîle-hata*
꼬부슴하다	*k'opusîm-hata*
꼬부장하다	*k'opucang-hat*
꼬부장꼬부장하다	*k'opucangk'opucang-hata*
꼬불꼬불하다	*k'opulk'opul-hata*
꼬불탕하다	*k'opulthang-hata*
꼬불탕꼬불탕하다	*k'opulthangk'opulthang-hata*
꼬불퉁하다	*k'opulthong-hata*
꼬붓하다	*k'opus-hata*
꼬붓꼬붓하다	*k'opusk'opus-hata*
구부렁하다	*kupulông-hata*
구부렁구부렁하다	*kupulôngkupulông-hata*
구부스름하다	*kupusîlîm-hata*
구부스레하다	*kupusîle-hata*
구부슴하다	*kupusîm-hata*
구부정하다	*kupucông-hata*
구부정구부정하다	*kupucôngkupucông-hata*
구불구불하다	*kupulkupul-hata*
구불텅하다	*kupulthông-hata*
구불텅구불텅하다	*kupulthôngkupulthông-hata*
구불퉁하다	*kupulthung-hata*
구붓하다	*kupus-hata*
구붓구붓하다	*kupuskupus-hata*
꾸부렁하다	*k'upulông-hata*
꾸부렁꾸부렁하다	*k'upulôngk'upulông-hata*
꾸부스름하다	*k'upusîlîm-hata*
꾸부스레하다	*k'upusîle-hata*
꾸부슴하다	*k'upusîm-hata*
꾸부정하다	*k'upucông-hata*
꾸부정꾸부정하다	*k'upucôngk'upucông-hata*
꾸불꾸불하다	*k'upulk'upul-hata*
꾸불텅하다	*k'upulthông-hata*
꾸불텅꾸불텅하다	*k'upulthôngk'upulthông-hata*
꾸불퉁하다	*k'upulthung-hata*
꾸붓하다	*k'upus-hata*
꾸붓꾸붓하다	*k'upusk'upus-hata*

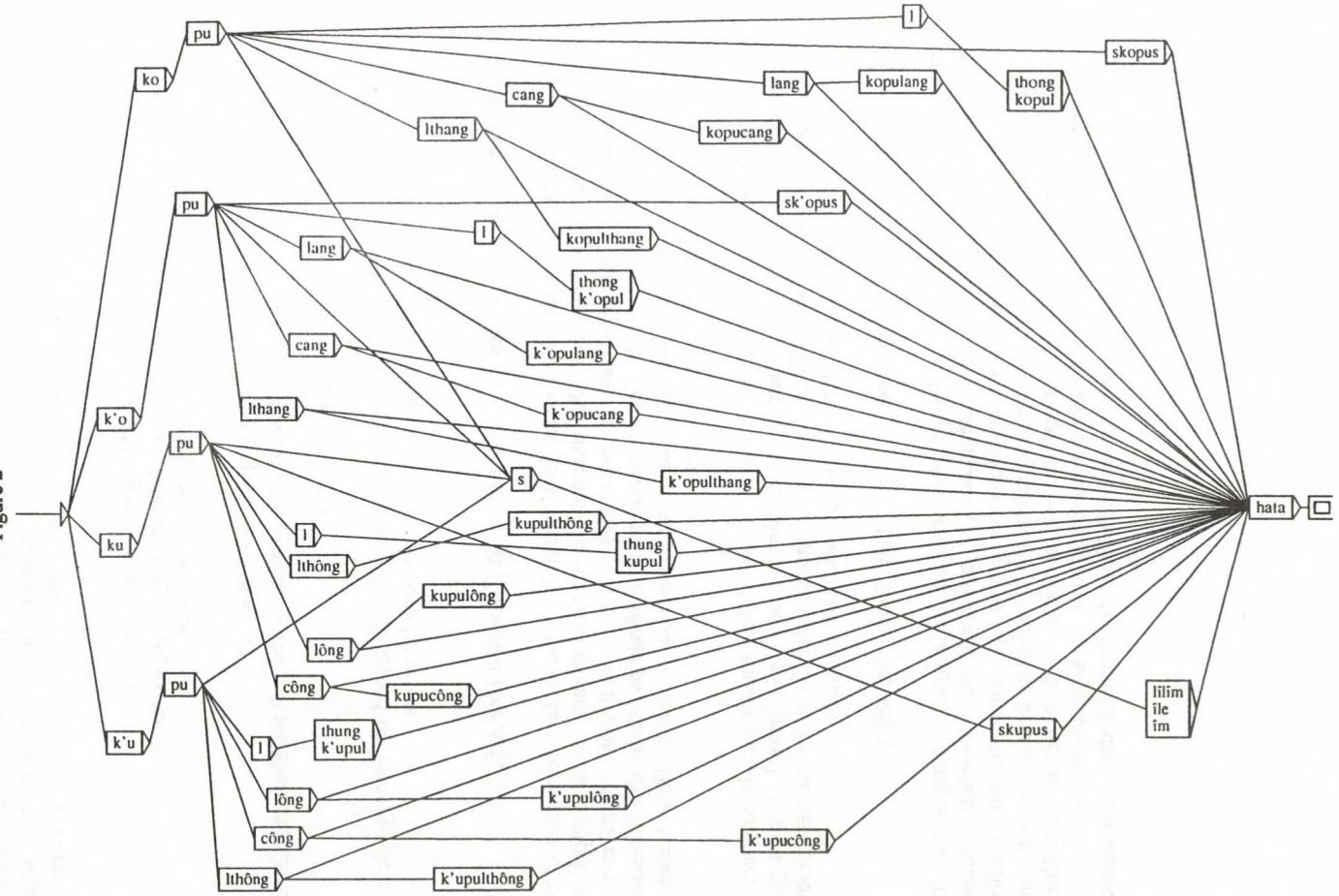
Ce processus est fort productif. La combinatoire des variantes dans (1) est la suivante :

(2) *{(ko + k'o)-pu-(lang + cang + lthang + s)}$^{(1+2)}$ -hata*
(ko + k'o)-pu-(sîlîm + sîle + sîm + lthong)-hata
{(ko + k'o + ku + k'u)-pul}^{2} -hata
{(ku + k'u)-pu-(lông + công + lthông + s)}$^{(1+2)}$ -hata
(ku + k'u)-pu-(sîlîm + sîle + sîm + lthung)-hata

(L'exposant 2 indique que chacune des séquences correspondant au contenu de l'accolade peut être répétée (e.g. *kopulangkopulanghata*).)

Cette combinatoire pourra être présentée sous la forme d'un graphe d'automate fini. Le graphe de la figure 2 représente 52 variantes consonantiques et vocaliques pour l'adjectif [courbe].

Figure 2



/namjs/courbe

Ce graphe pourra être subdivisé en fonction des propriétés syntaxiques qui caractérisent ces variantes : par exemple, dans la série (1), certains termes prennent comme sujet un substantif *humain* (ou un substantif *métonymique* de type "substantif de partie du corps"), d'autres non. Soit par exemple :

(3) 할머니의 허리가 (꼬부랑하 + *구불텅구불텅하 + 구부정하)-다
[halmôni **vieille dame** - *ii* **Gén** *hôli* **dos***]* - *ka* **nmtf** *(k'opulangha* **courbe** + **kupulthôngkupulthôngha* **courbe** + *kupucôngha* **courbe***)* - *ta* **St**
(Le dos de cette vieille dame est courbe)

(4) 저 산허리가 (*꼬부랑하 + 구불텅구불텅하 + *구부정하)-다
cô **ce** *sanhôli* **flanc de montagne** - *ka* **nmtf** *(*k'opulangha* **courbe** + *kupulthôngkupulthôngha* **courbe** + **kupucôngha* **courbe***)* - *ta* **St**
(Le flanc de cette montagne est courbe)

L'intérêt d'une telle représentation devient considérable quand on envisage de décrire les formes fléchies de ces adjectifs telles qu'on les trouve dans les textes.

1.2. Combinatoire des suffixes de l'expression [*Si (Nhum)0 être Adj, ...*]

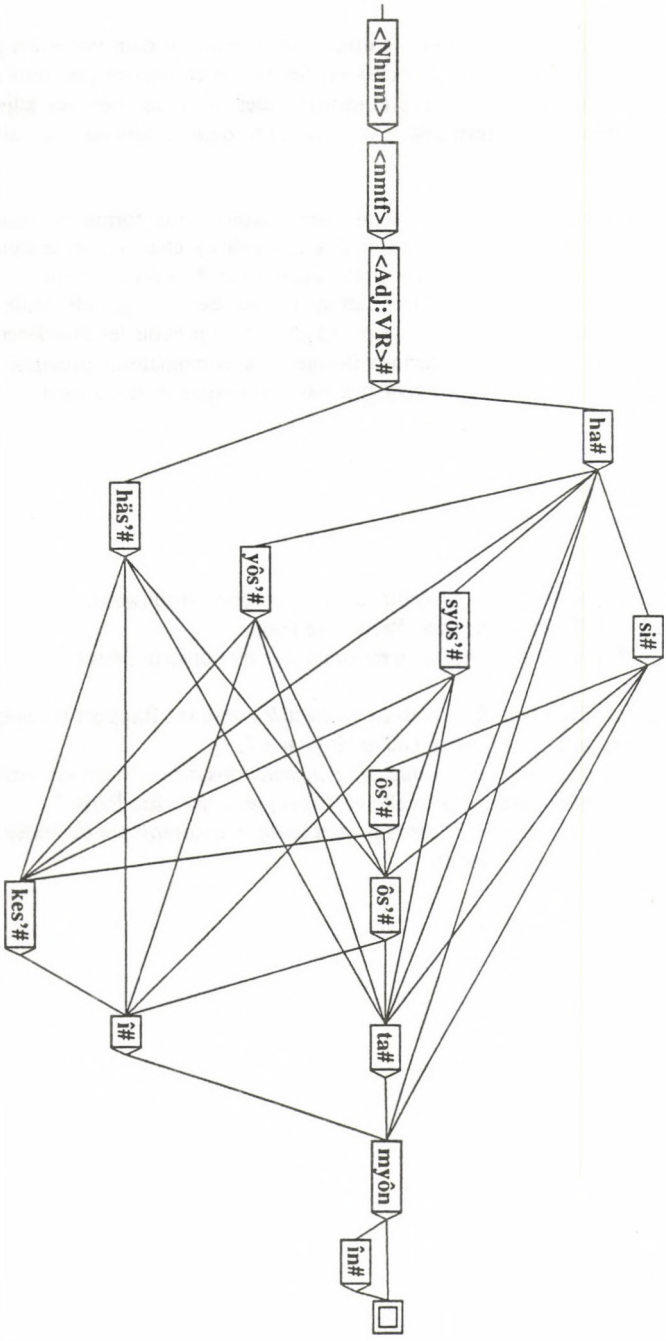
Soulignons que ce sont les formes fléchies que l'on oberve dans les textes et non la forme canonique (non-conjuguée). Des descriptions morphologiques, aussi complètes que possible sont alors indispensables et préalables à toute entreprise d'informatisation systématique des données lexicales. Cette situation ne nous révèle pas seulement le besoin d'un enregistrement exhaustif des formes fléchies mais aussi la nécessité d'une présentation adéquate pour éviter des données trop lourdes.

Considérons un cas. Le suffixe conjonctif *-myôn* (suffixe équivalent à la conjonction de subordination de condition "si" en français) choisi dans la case F de la figure 1, ne peut pas être employé après un suffixe d'honorification de l'interlocuteur (case E) et il n'est pas compatible avec le suffixe de temps futur. La combinatoire d'une séquence du type :

(5) *[(Nhum)0-nmtf <Adj:VR>-W-myôn],* ([Si (Nhum)0 être Adj],)

pourra être présentée sous la forme d'un graphe. Le graphe de la figure 3 comporte 50 formes fléchies de l'adjectif qui entre dans la séquence (5), autrement dit, il représente toutes les formes de subordonnées de **condition** introduite par *-myôn* (si), construites sur un adjectif à sujet *humain*. La série des adjectifs [courbe] comporte 10 adjectifs à sujet *humain*. On voit alors que le graphe de la figure 3 nous permettra de représenter 500 formes fléchies. Quand le nombre des variantes d'une série est élevé comme dans le cas des adjectifs de couleur [noir] (la centaine), on imagine facilement de quelle manière l'effectif des formes fléchies sera augmenté.

Figure 3



2. Perspectives

La combinatoire générale des suffixes autour d'un adjectif ou d'un verbe n'a jamais été indiquée explicitement dans les grammaires traditionnelles (et on comprend pourquoi). Il en est de même pour la combinatoire des variantes consonantiques et vocaliques des adjectifs. Par ailleurs, ce genre de série combinatoire s'observe également dans le lexique des adverbes ou des verbes.

Nous pourrons envisager de traiter cette combinatoire sous forme de liste globale, contenant toutes les formes fléchies, codées pour être associées à chaque forme canonique. Il nous semble que cette procédure n'est pas adéquate dans le cas du coréen, étant donné que la série des formes fléchies pour chaque forme canonique est de trop grande taille. Il serait préférable d'étudier séparément chaque série de paradigme et de prévoir les interférences entre ces séries, au lieu d'énumérer toutes les formes fléchies. La combinatoire prendrait la forme d'une grammaire locale morpho-syntaxique telle que celle du graphe de la figure 3.

Références

Chö, Hyôn-Pä, 1929 (réédit 1989), *Uli Malpon*, Séoul : Côngîm Munhwasa.

Gross, Maurice, 1975, *Méthodes en syntaxe*, Paris : Hermann.

Kim, Sûk-Tïk, 1992, *Ulimal Hyôngthälon* (La morphologie du coréen), Séoul : Thapchulphansa.

Nam, Jee-Sun, 1991, *Etablissement du corpus des adjectifs coréens* : Rapport technique N° 30, Paris : Institut Blaise Pascal, Université Paris 7.

Nam, Jee-Sun, 1994, *Classification syntaxique des constructions adjectivales en coréen*, thèse de doctorat en linguistique formelle et théorique, Université Paris 7.

Silberztein, Max, 1993, *Dictionnaires électroniques et analyse automatique de textes - le système INTEX*, Paris : Masson.

Annexe 1 : *Couleur noire*

가마노르께하다	*kamanolĭk'ehata*
가마무트름하다	*kamamuthĭlĭmhata*
가마반드르하다	*kamapantĭlĭhata*
가마반지르하다	*kamapancilĭhata*
가맣다	*kamahta*
가무끄름하다	*kamuk'ĭlĭmhata*
가무대대하다	*kamutătăhata*
가무댕댕하다	*kamutăngtănghata*
가무레하다	*kamulehata*
가무숙숙하다	*kamusuksukhata*
가무스럼하다	*kamusĭlŏmhata*
가무스레하다	*kamusĭlehata*
가무스름하다	*kamusĭlĭmhata*
가무잡잡하다	*kamucapcaphata*
가무족족하다	*kamucokcokhata*
가무칙칙하다	*kamuchikchikhata*
가무퇴퇴하다	*kamuthŏthŏhata*
가뭇가뭇하다	*kamuskamushata*
가뭇하다	*kamushata*
감노랗다	*kamnolahta*
감노르다	*kamnolĭta*
감다	*kamta*
감디감다	*kamtikamta*
감파랗다	*kamphalahta*
감파르다	*kamphalĭta*
감파르잡잡하다	*kamphalĭcapcaphata*
감파르족족하다	*kamphalĭcokcokhata*
거머누르께하다	*kŏmŏnulĭk'ehata*
거머무트룩하다	*kŏmŏmuthĭlukhata*
거머무트름하다	*kŏmŏmuthĭlĭmhata*
거머번드르하다	*kŏmŏpŏntĭlĭhata*
거머번지르하다	*kŏmŏpŏncilĭhata*
거머칙칙하다	*kŏmŏcikcikhata*
거머충충하다	*kŏmŏchungchunghata*
거멓다	*kŏmŏhta*
거무끄름하다	*kŏmuk'ĭlĭmhata*
거무대대하다	*kŏmutetehata*
거무뎅뎅하다	*kŏmutengtenghata*
거무레하다	*kŏmulehata*
거무숙숙하다	*kŏmusuksukhata*
거무스레하다	*kŏmusĭlehata*
거무스름하다	*kŏmusĭlĭmhata*
거무숙하다	*kŏmusĭkhata*
거무접접하다	*kŏmucŏpcŏphata*
거무죽죽하다	*kŏmucukcukhata*
거무축축하다	*kŏmuchukchukhata*
거무충충하다	*kŏmuchungchunghata*
거무칙칙하다	*kŏmuchikchikhata*
거무테테하다	*kŏmuthethehata*
거무퉈퉈하다	*kŏmuthwithwihata*
거무트름하다	*kŏmuthĭlĭmhata*
거무틱틱하다	*kŏmuthikthikhata*
거뭇거뭇하다	*kŏmuskŏmushata*
거뭇하다	*kŏmushata*
검누렇다	*kŏmnulŏhta*

검누르다	*kômnulîta*
검다	*kômta*
검디검다	*kômtikômta*
검붉다	*kômpulkta*
검뿌옇다	*kômp'uyôhta*
검퍼렇다	*kômphôlôhta*
검푸르다	*kômphulîta*
검푸르접접하다	*kômphulîcôpcôphata*
검푸르죽죽하다	*kômphulîcukcukhata*
까마무트름하다	*k'amamuthîlîmhata*
까마반드르하다	*k'amapantîlîhata*
까마반지르하다	*k'amapancilîhata*
까맣다	*k'amahta*
까무끄름하다	*k'amukîlîmhata*
까무대대하다	*k'amutătăhata*
까무댕댕하다	*k'amutăngtănghata*
까무레하다	*k'amulehata*
까무숙숙하다	*k'amusuksukhata*
까무스레하다	*k'amusîlehata*
까무스름하다	*k'amusîlîmhata*
까무잡잡하다	*k'amucapcaphata*
까무족족하다	*k'amucokcokhata*
까무촉촉하다	*k'amuchokchokhata*
까무충충하다	*k'amuchongchonghata*
까무칙칙하다	*k'amuchikchikhata*
까무퇴퇴하다	*k'amuthôthôhata*
까무트름하다	*k'amuthîlîmhata*
까뭇까뭇하다	*k'amusk'amushata*
까뭇하다	*k'amushata*
깜다	*k'amta*
꺼머무트름하다	*k'ômômuthîlîmhata*
꺼머번드르하다	*k'ômôpôntîlîhata*
꺼머번지르하다	*k'ômôpôncilîhata*
꺼멓다	*k'ômôhta*
꺼무끄름하다	*k'ômukîlîmhata*
꺼무대대하다	*k'ômutetehata*
꺼무댕댕하다	*k'ômutengtenghata*
꺼무레하다	*k'ômulehata*
꺼무스레하다	*k'ômusîlehata*
꺼무스름하다	*k'ômusîlîmhata*
꺼무숙하다	*k'ômusîkhata*
꺼무접접하다	*k'ômucôpcôphata*
꺼무죽죽하다	*k'ômucukcukhata*
꺼무촉촉하다	*k'ômuchukchukhata*
꺼무충충하다	*k'ômuchungchunghata*
꺼무칙칙하다	*k'ômuchikchikhata*
꺼무테테하다	*k'ômuthethehata*
꺼무퉈퉈하다	*k'ômuthwithwihata*
꺼무트름하다	*k'ômuthîlîmhata*
꺼뭇꺼뭇하다	*k'ômusk'ômushata*
꺼뭇하다	*k'ômushata*
껌다	*k'ômta*
껌적껌적하다	*k'ômcôkk'ômcôkhata*

Annexe 2 : Echantillon des formes fléchies de l'adjectif *pulhănghata* (malheureux)

불행하거나	*pulhăngha-kôna*
불행하거늘	*pulhăngha-kônil*
불행하거니	*pulhăngha-kôni*
불행하거니와	*pulhăngha-kôniwa*
불행하거든	*pulhăngha-kôtîn*
불행하건만	*pulhăngha-kôn-man*
불행하게	*pulhăngha-ke*
불행하게는	*pulhăngha-ke-nîn*
불행하게도	*pulhăngha-ke-to*
불행하게만	*pulhăngha-ke-man*
불행하게만은	*pulhăngha-ke-man-în*
불행하겠거나	*pulhăngha-kes'-kôna*
불행하겠거늘	*pulhăngha-kes'-kônil*
불행하겠거니	*pulhăngha-kes'-kôni*
불행하겠거니와	*pulhăngha-kes'-kôniwa*
불행하겠거든	*pulhăngha-kes'-kôtîn*
불행하겠건만	*pulhăngha-kes'-kôn-man*
불행하겠고	*pulhăngha-kes'-ko*
불행하겠고도	*pulhăngha-kes'-ko-to*
불행하겠기는	*pulhăngha-kes'-kinîn*
불행하겠기는커녕	*pulhăngha-kes'-kinînkhônyông*
불행하겠기 때문에	*pulhăngha-kes'-kit'âmune*
불행하겠기로	*pulhăngha-kes'-kilo*
불행하겠기로서니	*pulhăngha-kes'-kilosôni*
불행하겠기를	*pulhăngha-kes'-ki-lîl*
불행하겠기만을	*pulhăngha-kes'-ki-man-îl*
불행하겠기에	*pulhăngha-kes'-ki-e*
불행하겠나	*pulhăngha-kes'-nya*
불행하겠나?	*pulhăngha-kes'-nya?*
불행하겠나가	*pulhăngha-kes'-nya-ka*
불행하겠나고	*pulhăngha-kes'-nya-ko*
불행하겠나고는	*pulhăngha-kes'-nya-ko-nîn*
불행하겠나고도	*pulhăngha-kes'-nya-ko-to*
불행하겠나고만	*pulhăngha-kes'-nya-ko-man*
불행하겠나고만은	*pulhăngha-kes'-nya-ko-man-în*
불행하겠나는	*pulhăngha-kes'-nya-nîn*
불행하겠나도	*pulhăngha-kes'-nya-to*
불행하겠나를	*pulhăngha-kes'-nya-lîl*
불행하겠나만	*pulhăngha-kes'-nya-man*
불행하겠나만은	*pulhăngha-kes'-nya-man-în*
불행하겠나만을	*pulhăngha-kes'-nya-man-îl*
불행하겠나만이	*pulhăngha-kes'-nya-man-i*
불행하겠나에따라	*pulhăngha-kes'-nya-et'ala*
불행하겠느냐	*pulhăngha-kes'-nî-nya*
불행하겠느냐가	*pulhăngha-kes'-nî-nya-ka*
불행하겠느냐고는	*pulhăngha-kes'-nî-nya-ko*
불행하겠느냐고도	*pulhăngha-kes'-nî-nya-ko-nîn*
불행하겠느냐고만	*pulhăngha-kes'-nî-nya-ko-to*
불행하겠느냐고만은	*pulhăngha-kes'-nî-nya-ko-man*
불행하겠느냐는	*pulhăngha-kes'-nî-nya-ko-man-în*
불행하겠느냐도	*pulhăngha-kes'-nî-nya-nîn*
불행하겠느냐를	*pulhăngha-kes'-nî-nya-to*
불행하겠느냐만	*pulhăngha-kes'-nî-nya-lîl*
불행하겠느냐만은	*pulhăngha-kes'-nî-nya-man*
불행하겠느냐만을	*pulhăngha-kes'-nî-nya-man-în*
불행하겠느냐만이	*pulhăngha-kes'-nî-nya-man-îl*
불행하겠느냐에따라	*pulhăngha-kes'-nî-nya-man-i*
불행하겠는데	*pulhăngha-kes'-nî-nya-et'ala*
불행하겠는데도	*pulhăngha-kes'-nînte*
불행하겠는지	*pulhăngha-kes'-nînte-to*
불행하겠는지는	*pulhăngha-kes'-nîn-ci*
불행하겠는지도	*pulhăngha-kes'-nîn-ci-nîn*
	pulhăngha-kes'-nîn-ci-to
불행하겠는지만	*pulhăngha-kes'-nîn-ci-man*
불행하겠다	*pulhăngha-kes'-ta*
불행하겠다거니와	*pulhăngha-kes'-ta-kôniwa*
불행하겠다고	*pulhăngha-kes'-ta-ko*
불행하겠다고는	*pulhăngha-kes'-ta-ko-nîn*
불행하겠다고도	*pulhăngha-kes'-ta-ko-to*
불행하겠다고만	*pulhăngha-kes'-ta-ko-man*
불행하겠다고만은	*pulhăngha-kes'-ta-ko-man-în*
불행하겠다기보다	*pulhăngha-kes'-ta-kipota*
불행하겠다기에	*pulhăngha-kes'-ta-ki-e*
불행하겠다는	*pulhăngha-kes'-ta-nîn*
불행하겠다니	*pulhăngha-kes'-ta-ni*
불행하겠다더니	*pulhăngha-kes'-ta-tô-ni*
불행하겠다더니만	*pulhăngha-kes'-ta-tô-ni-man*
불행하겠다더니만은	*pulhăngha-kes'-ta-tô-ni-man-în*
불행하겠더라도	*pulhăngha-kes'-tôlato*
불행하겠습니까?	*pulhăngha-kes'-sîpni-k'a?*
불행하겠습니다	*pulhăngha-kes'-sîpni-ta*
불행하겠으나	*pulhăngha-kes'-îna*
불행하겠으니까	*pulhăngha-kes'-înik'a*
불행하겠으면	*pulhăngha-kes'-îmyôn*
불행하겠으면은	*pulhăngha-kes'-îmyôn-în*
불행하겠음만을	*pulhăngha-kes'-îm-man-în*
불행하겠음은	*pulhăngha-kes'-îm-man-îl*
불행하겠음을	*pulhăngha-kes'-îm-în*
불행하겠음의	*pulhăngha-kes'-îm-îl*
불행하겠음이	*pulhăngha-kes'-îm-îi*
불행하겠지만	*pulhăngha-kes'-îm-i*
불행하겠지만도	*pulhăngha-kes'-ciman*
불행하겠지만은	*pulhăngha-kes'-ciman-to*
불행하고	*pulhăngha-kes'-ciman-în*
불행하고도	*pulhăngha-ko*
불행하기	*pulhăngha-ko-to*
불행하기가	*pulhăngha-ki*
불행하기는	*pulhăngha-ki-ka*
불행하기는커녕	*pulhăngha-ki-nîn*
불행하기 때문에	*pulhăngha-ki-nîn-khônyông*
불행하기로	*pulhăngha-ki-t'âmune*
불행하기로서니	*pulhăngha-kilo*
불행하기를	*pulhăngha-kilosôni*
불행하기만으로	*pulhăngha-ki-lîl*
불행하기만으로도	*pulhăngha-ki-manîlo*
불행하기만으로써도	*pulhăngha-ki-manîlo-to*
불행하기만을	*pulhăngha-ki-manîlos'ô-to*
불행하기만이	*pulhăngha-ki-man-îl*
불행하기보다	*pulhăngha-ki-man-i*
불행하기에	*pulhăngha-ki-pota*
불행하길래	*pulhăngha-ki-e*
불행하나	*pulhăngha-kilă*
불행하나마	*pulhăngha-na*
불행하나?	*pulhăngha-nama*
불행하나가	*pulhăngha-nya?*
불행하나고	*pulhăngha-nya-ka*
불행하나고는	*pulhăngha-nya-ko*
불행하나고도	*pulhăngha-nya-ko-nîn*
불행하나고만	*pulhăngha-nya-ko-to*
불행하나고만은	*pulhăngha-nya-ko-man*
불행하나는	*pulhăngha-nya-ko-man-în*
불행하나도	*pulhăngha-nya-nîn*
불행하나를	*pulhăngha-nya-to*
불행하나만	*pulhăngha-nya-lîl*
불행하나만은	*pulhăngha-nya-man*
불행하나만을	*pulhăngha-nya-man-în*
불행하나만이	*pulhăngha-nya-man-îl*
불행하나에따라	*pulhăngha-nya-man-i*
	pulhăngha-nya-et'ala

Project Report on the Historical Dictionary of Hungarian

JÚLIA PAJZS

The project for compiling the Historical Dictionary of Hungarian started nine years ago. This paper presents the current developments of our work.

As we have reported at the earlier COMPLEX conference, our department is working on compiling the Historical Dictionary of Hungarian by using a computerized corpus. We are still enlarging the corpus, at the moment we have more than 16 million running words on-line. In the present paper I would like to demonstrate the sofware tools we are using for text retrieval and editing the dictionary.

1. Text retrieval

For retrieving our corpus we use the PAT program (GONNET - TOMPA 1987, SALMINEN - TOMPA 1992) which was demonstrated by Professor TOMPA at our last conference. Because we find it very useful for our task, I will show you some of the possibilities offered by this software.

It has two kinds of use under UNIX: the normal "character mode", and PatMotif, which runs under XWindows. Both modes have their advantages and disadvantages, so we use them alternately depending on the task at hand.

Our corpus is lemmatized by a morphological analyser program, (Prószéky and Tihanyi 1992). This program was used for tagging the 19th-20th century corpus, so now we can retrieve the lexemes instead of just the running words. By using the codes, we can also make several interesting investigation: we can find the words according to part of speech, we search suffixes, and combination of suffixes etc. Sometimes, however we retrieve the original running text too, because the analyzed text can be sometimes missleading. (The analysis is made automatically, and the output is not corrected manually, therefore several mistakes and unresolved disambiguities remained in it.)

We use SGML-like symbols for recording the bibliographic data, and we use a combination of letters and digits for representing the Hungarian characters with diacritics. (Our UNIX workstation is an AT 486 compatible with SCO UNIX, this version is not able to handle the accented characters on the terminals.) A sample from our lemmatized text is in *Fig. 1*

```
<section>
<id>2000284001</id>
<author>JUHA1SZ GYULA</author>
<title>MINEK SZALADTOK EL...</title>
<publ>BUDAPEST; AKADE1MIAI; 1963;</publ>
<lform>2</lform>
<wdate>1906</wdate>
<text><page><p>0089</p>
 #/#    Minek[HA]    szalad[IGE]+tok[t2]    el[IK]    ti[NM]
sze1p[MN]+ek[PL]? / Te[NM] pillanat[FN], te[NM] a1lom[FN], te[NM]
```

```
e1let[FN]&,
/ Minek[HA] szalad[IGE]+tok[t2] el[IK] ti[NM]? #/ Minek[HA]
oly[NM] sze1p[MN] a[DET] la1ny[FN], az[DET]& a1lom[FN], /
Hogy[KOT] fe1l[IGE]+ek[e1] to3le[HA], ha[KOT]&
meg[IK]+tala1l[IGE]+om[Te1], / Hogy[KOT] nem[HA]& mer[IGE]+ek[e1]
szeret[IGE]+ni[INF]? #/ Minek[HA] fa1j[IGE]+t[Me3]& oly[NM]
nagyon[HA]& a[DET] le1lek[FN]=lelk+em[PSe1], / Mikor[HA] a[DET]
leg[FF]+sze1p[MN]=sze+bb[FOK] la1ny[FN]+ra[SUB]
le1[IGE]+tem[TMe1]&, / S[KOT] me1rt[HA]& nem[HA]& mer[IGE]+ek[e1]
feled[IGE]+ni[INF]? #/ Me1rt[HA]& von[IGE] fele1[NU]& hiu1[MN],
vad[MN] a1ram[FN]&, / Hogy[KOT] meg[IK]+tala1l[IGE]+jam[TPe1],
hogy[KOT] meg[IK]+ima1d[IGE]+jam[TPe1] / S[KOT] ne[HA]
mer[IGE]+jem[TPe1] meg[IK]+szeret[IGE]+ni[INF]? #/#
</text></section>
```

Fig. 1.

1.1 PAT

We can easily search any lexeme and it is fairly simple to modify the size of the context to be printed. *(Fig. 2)*

```
>> hamis
>> pr sample
   222843, ..Ama' gyilkos, hamis hi1rt! Mint O3szszel a'
li1liom-sza1l, / Sze..
   229789, .. hogy o3 olly hamis; / De tu3ri: mert nagy kedve1ben
/ Van o3 sz..
   284204, ..ssz, Valo1 's Hamis ko2zo2tt, - / Lelku2nk' fele1t,
Szi1vu2nk' f..
   373776, ..kra, mind a' hamissakra. (9 Matth. 5. v. 45. )9
</par><par> (1..
   619151, ..gy maga ellen hamis hitet esku2djo2n, e1s
mindekkora1ig be1kesse..
   692890, ..gokat, az az: hamis esku2ve1so2ket. Oskolai letzke1m
uta1n Majer..
   710376, .. egyenes, nem hamis, szive1vel egyu2tt, mio1ta tsak
vele ta1rsol..
   619151, ..gy maga ellen hamis hitet esku2djo2n, e1s
mindekkora1ig be1kesse1ges..

>> {PrintLength 215}
>> pr sample
   692890, ..gokat, az az: hamis esku2ve1so2ket. Oskolai letzke1m
uta1n Majer Jo1zsef Sze1kesfehe1rvari Kispap Bara1tomhoz tu2zes
levelet ke1szi1tettem. Hozza1ja fu2ggesztettem sza1ma1ra Ne1ha1ny
okaimat, Vitkovitsnak egy pa1r Halotti kes..
   710376, .. egyenes, nem hamis, szive1vel egyu2tt, mio1ta tsak
vele ta1rsolkodom, mindenkor bo2tsben volt elo3ttem: most ke1t
szinu3 keszkeno3se1get nem o2smero3 hi1v bara1tsa1ga ege1szen
szerteszte1t el terjedo3 bizodalom gyo2keret term..
  1435700, ..usonn, e1s a' hamis hithez tartozo1konn."
</par><par>,,Hija1ban szegezne1 maga1t ellene egyfelo24l a'
To2ro2k, masfelo24l a' Maurus; mert o24 tu11 az Eufratesenn, tu11
a' Taurus' havas be1rtzeinn, e1s tu11 vihetne1 azokonn az..
  1566571, ..tudoma1nyt a' hamisto11 megku2lo2mbo2zteti a'
to2ke1lletes okoskoda1s. Aka1r elo24bb, aka1r uto1bb, tsak
kitets43zik ve1gte1re, mint a' s43zeg a' zsa1kbo11, a' gonos43z
```

```
to2rekede1s, a' tsalfa foga1s. </par><par>(1 A' magya..
>> {LeftContext 100}
>> pr sample
   619151, ..bal ko2zt megtanultam  ku2lo2nbse1get tenni, rea1
nem vehettem le1kem esme1rete1t,  hogy maga ellen hamis hitet
esku2djo2n, e1s mindekkora1ig be1kesse1ges  tu2re1ssel tartottam
sza1mot a Mindenhato1nak gondvisele1se1re,  ki a..
```

Fig. 2.

It is possible to see some specifier data at the beginning of the context line, for example the data of writing, or name of the author. *(Fig. 3.)*

```
>> {SortOrder Occurhead wdate section}
>> a111
>> pr
1802          ..43zs43zen s43zeme fe4nynye? hogy s43enki  s43e
la1sson keres43ztu241 a' vontt ha11yogon? 's hogy ki-a111hass43a
ez  a'  mester  foga1s  me1g  ama'  s43zent  buzgo1sa1gu1,  's
elo24re-is,  ha1tra-is  la1to1 Jojada1nak s43zeme' e11e1t?..
1802          .. A' Pla1ne1ta1k ko2zzu241  ala1-hengeredett, /
Do2rge1se1nek hangja1t E1s43zak-fele1 tartya:  / Ho1 a111 a'
Ba11ticom, 's feje1r tenger' partja; / Mellyre fel-rettenve1n
a' Fennai kebel, / Mintegy mada1r s43zava eso24zve1n v..
1802        .. / Bara1zda11ta1k habos si1kja1t haj o1 s43za111a1;
/ Haddu1  a' vitorla1k' abros43zs43zai ala1 / Ki-a111va1n;
indu11ttok' dalla1t  fujdogala1. / Ku2rtje' hangza1sa1ra a'
s43ze11 ki-rohane1k; / A' ko1rma1nyos  pedig ama' nagy ..
1802          ..3zedegett, / 'S maga1ban  a' mezo24 gyapja1t
beretva11lya; / A' nya1j-fele1 tarto1 o2sve1nnye1t  e1-a111ya,
/ A' s43za1ja1ban le1vo24 fu24ve1 azt meg-kapva1n, / Nyekego24
ge1ge1je1t hamar ki-harapva1n, / Le-nyu1zott tzi1mere..
1802              ..date> <text><page><p>0220</p><par>  A' ke1s
fintorodva  la1ba1val  fel-fordu11tt,  /  Ve1re  fu21e  fele1
a111-kaptza1ja1n tsordu11tt; / Ha1tu11ro11 ki-futva1n Szakada1r
s43em  ke1se, / Hamar ki ra1ntatva1n Zemefrisnek ke1se, / ,,H..
1802       ..m ve1ge1t itt  le1ve1n ves43ztemre, / ,,Ba1torsa1god
felo241  de  me1g-is  fel-tettem,  ,,Hogy  egy  vad-a111atnak
meg-fele1s43z helyettem; / ,,E1n ugyan nem  s43za1nna1m e11temet
e1retted; ,,De illy buzgo1sa1gom mos43tan fe11re  ..
1802          ..' Szu24z  lea1nyhoz  mikor ke1ro24k jo2nnek, / Es
s43zu2lo24i elo24tt azok be1-ko2s43zo2nnek,  / Meg a111 keze1ben
vo11t Rokka, vagy Moto11la / Keble1ben moto1zva1n,  tsak ke1so24n
meg-s43zo11la; / I1gy Zemefris melje alma1ja1..
1802        ..t e11u2nk, / Ha erko241tso2s s43zi1vtek egyet tart
mive11u2nk!  / Zemefris s43zava1ra Szakada1r ra1-a111a, / E1s
go24go2s  pitvaru1  ha1za1ba  be1-s43za111a.  </par> </page>
</text> </section>  <section> <id>1900325010</id> <au..
1802          .. es41zko2zo2ket a Tes41tbo241 fogyas41ztja1k,
ero24teleni1tto24  es41zko2zo2knek  tarthatjuk,  e1s  azt
a111i1thatjuk, hogy minden u2resede1s, le1gyen az hasmene1s,
ha1nya1s, izzada1s, ve1rfolya1s, nya11folya1s, s41zoptata1s,
no2..
```

Fig. 3.

Cooccurrences and "non cooccurences" of words, codes, suffixes can be searched as well. This is probably the most powerful tool for the different fields of research: now one can easily testify which suffix combinations occur in the real corpus. So far we only knew which are the rules of Hungarian morphology in principle (for example a Hungarian noun can have more than 600 different suffix combinations), now at least we can have experience on reality. We have found, not surprisingly, that the really complicated suffixations hardly occur at all. For example, you can have a form like this *apá[FN] +m[PSe1] +é[POS] +i[PL] +nak[DAT]* (which is the dative of possessivus pluralis of the genitive of the noun '*father*'), but in 4 million examples this form never occurs, and there are only 58 cases altogether when the genitive is followed by the possessive pluralis suffix. *(Fig. 4.)*

```
>> PS fby.10 POS
>> 320
>> pr
1969      ..], a[DET] csoport[FN] jo1sza1g[FN]+a1[PSe3]+t[ACC].
E1s[KOT] les[IGE]+d[Te2]& a[DET] bara1t[FN]+aid[PSe2i]+at[POS],
aki[NM]+k[PL]  ke1p[FN]+pel[INS]  se[HA]  fordul[IGE]+nak[t3]
fele1[HA]+d[PSe2]&. Apa[FN] vagy[KOT]& te[NM]? vic..
1974          ..gy[KOT] ez[NM] egy[DET]& isteni[MN] hecc[FN]?
</par><par>%Ja1tszd  a[DET]  kisded[FN]
ja1te1k[FN]+aid[PSe2i]+at[POS]  u1gy[HA],  ahogy[HA]
akar[IGE]+od[Te2]. E1rte[HA]+d[PSe2]&? </par><par>(1 George[FN]
(mi1mel[IGE]+t[Me3]& ba1mu..
1980KO2RU2  ..[FN]+a[PSe3]  kattog[IGE]+ja[Te3]&,  az[DET]&
egyedi[MN]+ne11[ADE]  is[KOT]  %egyedibb
gondolat[FN]+aid[PSe2i]+at[POS]:  #2  Ad[IGE]+j[Pe2]
u1r[FN]=Ur+am[PSe1] asszony[FN]+t[ACC], / egy[DET]& ma1r[HA]
van[IGE], de[KOT]& to2bbet[HA]..
>> PSe1i fby.10 pos
>> 1
>> PSe2i fby.10 pos
>> 44
>> PSe3i fby.10 pos
>> 12
>> PSt1i fby.10 pos
>> 4
>> PSt2i fby.10 pos
>> 0
>> PSt3i fby.10 pos
>> 0
```

Fig. 4

One can make any subset of a set of examples, and investigate them one by one. Any result can be saved during the session.

Using the SGML symbols as field markers we can retrieve the bibliographic data belonging to the quotation *(Fig. 5.)*. By the **region** and **within, including** commands, one can search in any specified fragment of the text (for example, the works of one author, etc.).

```
>> region section including [394933]
>> pr
  394046, .. <section> <id>1900325007</id> <author>PERETSE1NYI
NAGY  LA1SZLO1</author>  <title>SZAKADA1R'  ESTHONNYAI  MAGYAR
FEJEDELEM' BU1JDOSA1SA.</title> <publ>POZSONY-PEST; LANDERER
MIHA1LY;  1802;  SZAKADA1R'  ESTHONNYAI  MAGYAR  FEJEDELEM'
BU1JDOSA1SA.  A' </publ> <lform>1A1</lform> <wdate>1802</wdate>
```

```
<text><pag..
>> region page including [394933]
>> pr
   394341, ..><page><p>0220</p><par> A' kels fintorodva lalbalval
fel-fordulltt, / Velre fu2le felel alll-kaptzaljaln tsordulltt;
/ Haltullroll ki-futvaln Szakadalr s43em kelse, / Hamar ki
ralntatvaln Zemefrisnek kelse, / ,,Helyes43s43en, ilgy
szolllott, Zemefris As43zonyom! / ,,Eln azokra s43zinte zsidald
i..
```

Fig. 5.

With the **signif, rankedby** commands we can ask for some frequency data, we can learn what is the most frequent context of any searched string *(Fig. 6.)*.

```
>> signif PS fby.10 POS
>> 112, "PSel"
>> pr sample
  5031370,    ..]+m[PSe1]+e1[POS]+ban[INE]    %stalczlolt
tart[IGE]+ottam[TMe1]&    e1s[KOT]    e1n[NM]
Fiumelban^Fiul[FN]+m[PSe1]+e1[POS]+ban[INE]    nem[HA]&    a[DET]
tenger[FN]+t[ACC]& szeret[IGE]+em[Te1], - hiszen[KOT] nincs[IGE]
is[KOT] ott..
 56043953, ..edelly[FN], a[DET] szerelem[FN]& viviszekcio1[FN]:
a[DET] mals[NM]+e1[POS]&, a[DET] maga[NM]=maga+m[PSe1]+e1[POS].
- %Emerson szerint[NU] Goethe[FN]=Goethe1+t[ACC] egy-egy[SZN]
lelki[MN] vallsalg[FN], ulj[MN] gondol..
 44930843,    ..    nalla[HA]    a[DET]    mosoly[FN]+t[ACC]
jelent[IGE]+ette[TMe3]&.    </par><par>-    Az[DET]&
anya[FN]=anyal+m[PSe1]+e1[POS]    lehet[IGE]+ett[Me3]&    -
mond[IGE]+ta[TMe3]& velgu2l[HA]& is[KOT]. </par><par>- Hogy[KOT]
van[IGE] elde..

>> signif PS
>> 165574, "PSe3"
>> pr sample
 36885266, ..=szu2let+nek[t3] uljra[HA]&. A[DET] %havasalfo2ldi
vajda[FN], %Neagoe Basa[FN]+rab[FN] udvar[FN]+al[PSe3]+ba[ILL]
gyu3jt[IGE]+i[Te3] a[DET] Balkaln[FN]+roll[DEL] meneku2lo3[FN]
fo3pap[FN]+ok[PL]+at[ACC], ko2zt[HA]+u..
 56591286,    ..,,elvtalrs[FN]''-+nak[DAT],    hanem[KOT]
,,Fo3no2k[FN]&    ulr[FN]+nak[DAT]'',    els[KOT]
jelenlelt[FN]+el[PSe3]+ben[INE]&    mindenki[NM]+t[ACC]
elvtalrs[FN]+nak[DAT]    szollilt[IGE]+anelk[TFt3],    csak[HA]
Salr[FN]+i[IKEP] neln..
```

Fig. 6.

1.2 PatMotif

One can do the very same things in PatMotif, but not in the same way of course. It has some advantages compared to PAT: it is fairly easy to pick up one example and copy the desired part to another file. It is also more handy to search the corresponding bibliographic data and copy it to the other file.

It has, however, some disadvantages: you can only see a very little part of the context in the result area window, it is a little uncomfortable to look at them in the larger context one by one. It is also less usable for the lexemes which occurred frequently, you have to look

at each of them in the larger context and when you finish your session and you would like to continue it the next time, you can not just go to the last example, you have to start again from the beginning of the sample set.

This version is also less reliable - at least on our computer system - the session often halts during work.

2. The software for dictionary writing

We have a very handy tool for writing dictionary entries in SGML format: the WriterStation from Datalogics Limited. This software is a text editor for SGML documents. You have to develop an application for your task, which means that you have to write the DTD of your document (the dictionary in this case), then you have to specify what should happen to the different parts of the structure. From then on the system continuously notifies you which elements can be put in any level of the structure.

```
<!DOCTYPE dic
[
<!ELEMENT dic  o o  (art+)                                          >
<!ELEMENT art  - -  (hea, (grb*|sen+))                              >
<!ELEMENT hea  - -  (lem, hmn?, pos?, sug*, var*, xrf?)             >
<!ELEMENT sen  - -  (snu?, def+, sug*, xpl+, xrf?, sen*)            >
<!ELEMENT xpl  - -  (exa, pub)                                      >
<!ELEMENT pub  - -  (dpb, src?, aut, ttl, pag)                      >
<!ELEMENT src  - -  (#PCDATA)                                       >
<!ELEMENT grb  - -  (gnu, pos, sug*, var*, xrf?, sen+)              >
<!ELEMENT lem  - -  (#PCDATA)    -- lexem --                        >
<!ELEMENT hmn  - -  (#PCDATA)    -- homonim num      --             >
<!ELEMENT pos  - -  (#PCDATA)    -- part of speech --               >
<!ELEMENT sug  - -  (#PCDATA)    -- suggestion --                   >
<!ELEMENT var  - -  (#PCDATA)    -- variant --                      >
<!ELEMENT xrf  - -  (#PCDATA)    -- xref --                         >
<!ELEMENT snu  - -  (#PCDATA)    -- sense num--                     >
<!ELEMENT gnu   - - (#PCDATA)    -- gram. block num. --             >
<!ELEMENT def  - -  (#PCDATA)    -- definition --                   >
<!ELEMENT exa  - -  (#PCDATA)    -- example --                      >
<!ELEMENT dpb  - -  (#PCDATA)    -- date of pub --                  >
<!ELEMENT aut  - -  (#PCDATA)    -- author --                       >
<!ELEMENT ttl  - -  (#PCDATA)    -- title --                        >
<!ELEMENT pag  - -  (#PCDATA)    -- page --                         >
]>
```

Fig. 7.

The suggested DTD of our dictionary is in *(Fig. 7.)*. This is just a temporary version for the first draft entries, we will modify it until we decide on the terminal format.

When you use the editor, you do not have to write the SGML tags yourself, you simply have to use a key combination to enter it. The program shows the number of key and the name of the corresponding tag in the information window at the bottom of the screen. Here you can also see the text in SGML format, while in the text window the formatted entry can be read. If you mark the different typefaces with different colours your entry will be quite readable, similarly to normal text editors.

It is also possible to specify the typefaces for your printer, but unfortunatelly this part of the software is not usable for us, because it is not able to handle the accented characters. Therefore we developed another software for converting the SGML files to WordPerfect

files, where we replace the SGML tags for typeface markers. *Fig 8.* shows two draft entries in SGML format and in printed format.

```
<DIC>
<ART><HEA><LEM>alabástromfehér</LEM>  <POS>mn</POS>
<VAR>alabastromfehér</VAR>  </HEA><SEN><DEF>alabástromhoz
hasonlóan szép fehér színű</DEF> <XPL><EXA>igen sovány nyaka volt
és hosszú; de különben gyönyörű bájos szemekkel, szép
alabástromfehér fogakkal</EXA>  <PUB><DPB>1872/1955</DPB>
<AUT>Déryné</AUT>  <TTL>Eml. 1:</TTL>  <PAG>63</PAG></PUB>
</XPL><XPL><EXA>... az alabastrom fehér arcz ... szeliden néz le
reánk</EXA> <PUB><DPB>1884</DPB> <SRC>#</SRC> <AUT>Jókai</AUT>
<TTL>MagyFöld.</TTL> <PAG>56</PAG></PUB></XPL></SEN></ART>

<ART><HEA><LEM>alantas</LEM></HEA>
<GRB><GNU>I.</GNU> <POS>mn</POS> <SEN><DEF>alacsony, alacsonyan
levő, fekvő</DEF> <XPL><EXA>... a vele való folytonos érintkezés
s együtt tanulás csakis jótékony befolyással lehetett mi reánk
gyengébbekre s a fejlettség sokkal alantasabb fokán állókra
nézve</EXA>  <PUB><DPB>1824-1844/1887.</DPB>
<AUT>Podmaniczky</AUT>  <TTL>Napl.</TTL>
<PAG>128</PAG></PUB></XPL> <XPL><EXA>...ezek a bástyatornyok még
most is fenyegetnék szakállas ágyúikkal az alantas síkságot</EXA>
<PUB>  <DPB>1882/1897</DPB><SRC>#</SRC>  <AUT>Jókai</AUT>
<TTL>68:</TTL>  <PAG>281</PAG></PUB></XPL></SEN>
<SEN><SNU>2.</SNU><DEF>alacsony társadalmi állású, hivatali
beosztású</DEF> <XPL><EXA>A vendéglő kertjében, mely a promenád
felé terült el a lejtőn, szerényen leült a tipegő polgármester,
és egy ideig mélázva nézte a kugliző társaságot, a zöldhajtókás
tiszteket, a fürge alantas tisztviselőket...  </EXA>
<PUB><DPB>1893/1957</DPB>  <AUT>PappD</AUT>  <TTL>Muzs.</TTL>
<PAG>58</PAG></PUB></XPL>.  <XPL><EXA>A történet abban a
nagyvárosban játszódik, ahol szeretni és dalolni csupán az
alantas néposztályban szokás</EXA>  <PUB><DPB>1913</DPB>
<SRC>#</SRC>  <AUT>Krúdy</AUT>  <TTL>Postak.</TTL>
<PAG>6</PAG></PUB></XPL></SEN><SEN><SNU>3</SNU>
<DEF>alacsonyrendű, közönséges</DEF> <XPL><EXA>A hiba az én
kicsinyes, alantas lekemben rejlett</EXA> <PUB><DPB>1893</DPB>
<SRC>#</SRC><AUT>Kabos</AUT>  <TTL>Éjsz.</TTL>
<PAG>13</PAG></PUB></XPL> <XPL><EXA>S ez a mozdulata, ez a
magatartása olyasmit fejezett ki, hogy ugyan mi köze is van néki
az ilyen alantas dolgokhoz, mint terhesség, gyerekszülés?</EXA>
<PUB><DPB>1961</DPB>  <AUT>Füst M</AUT>  <TTL>Parn</TTL>
<PAG>156</PAG></PUB></XPL></SEN>
</GRB>
<GRB><GNU>II.</GNU> <POS>fn</POS> <SEN><DEF>alárendelt, beosztott
személy</DEF> <XPL><EXA>Valahányszor a nagyurak valami galyibába
keverednek, annak a levét az alantasok isszák meg </EXA>
<PUB><DPB>1910/1911</DPB>  <SRC>#</SRC>  <AUT>Mikszáth</AUT>
<TTL>Fekvár. 1:</TTL> <PAG>81</PAG></PUB></XPL> <XPL><EXA>És
mindenkivel beszélt és mindenkit irányított. Munkatársai,
alantasai el voltak ragadtatva tőle</EXA> <PUB><DPB>1922</DPB>
<AUT>Karinthy</AUT>  <TTL>Cap.</TTL>
<PAG>49</PAG></PUB></XPL></SEN></GRB></ART>
```

alabástromfehér *mn alabastromfehér* 'alabástromhoz hasonlóan szép fehér színű' *igen sovány nyaka volt és hosszú; de különben gyönyörű bájos szemekkel, szép alabástromfehér fogakkal* 1872/1955 Déryné Eml. 1:63... *az alabastrom fehér arcz ... szeliden néz le reánk* 1884 # Jókai MagyFöld.:56

alantas

I. *mn* **1.**'alacsony, alacsonyan levő, fekvő' ... *a vele való folytonos érintkezés s együtt tanulás csakis jótékony befolyással lehetett mi reánk gyengébbekre s a fejlettség sokkal alantasabb fokán állókra nézve* 1824-1844/1887. Podmaniczky Napl.:128 ...*ezek a bástyatornyok még most is fenyegetnék szakállas ágyúikkal az alantas síkságot* 1882/1897 # Jókai 68:281 **2.** 'alacsony társadalmi állású, hivatali beosztású' *A vendéglő kertjében, mely a promenád felé terült el a lejtőn, szerényen leült a tipegő polgármester, és egy ideig mélázva nézte a kuglizó társaságot, a zöldhajtókás tiszteket, a fürge alantas tisztviselőket...* 1893/1957 PappD Muzs.:58 *A történet abban a nagyvárosban játszódik, ahol szeretni és dalolni csupán az alantas néposztályban szokás* 1913 # Krúdy Postak.:6. **3.** 'alacsonyrendű, közönséges' *A hiba az én kicsinyes, alantas lekemben rejlett* 1893 # Kabos Éjsz.:13. *S ez a mozdulata, ez a magatartása olyasmit fejezett ki, hogy ugyan mi köze is van néki az ilyen alantas dolgokhoz, mint terhesség, gyerekszülés?* 1961 Füst M Parn.:156.

II. *fn* 'alárendelt, beosztott személy' *Valahányszor a nagyurak valami galyibába keverednek, annak a levét az alantasok isszák meg* 1910/1911 # Mikszáth Fekvár. 1:81. *És mindenkivel beszélt és mindenkit irányított. Munkatársai, alantasai el voltak ragadtatva tőle* 1922 Karinthy Cap.:49.

Fig. 8.

3. Concluding remarks

With most of our future corpus on-line and with these software tools we can at last start to work on the compilation of our dictionary. When most of the dictionary is on-line, we will be able to retrieve it by PAT also.

We would like to make available our corpus for any researchers working in the field of Hungarian linqustics and literature, and further develop it towards a Hungarian National Corpus.

References:

GONNET, G.: (1987) *PAT - An efficient text searching system.* University of Waterloo Centre for the New OED.

GONNET, G. - TOMPA, F.: (1987) *Mind your Grammar: A New Approach to Modelling Text.* University of Waterloo Centre for the New OED.

PAJZS, J.: (1990) Creating a Historical Dictionary of Hungarian with the Aid of Computer *T. MAGAY - J. ZIGÁNY: BUDALEX '88 Proceedings.* Budapest: Akadémiai Kiadó, pp. 559-563.

PAJZS, J.: (1991) The Use of a Lemmatized Corpus for Compiling the Dictionary of

Hungarian In: *Using Corpora Proceedings of the 7th Annual Conference of the OUP & Centre for the New OED and Text Research.* University of Waterloo Centre for the New OED, pp. 129-136.

PAJZS, J. - TIHANYI, L. - VILLÓ, I.: (1992) Writing Dictionaries with Grammar Defined Databases. In: *Papers on Computational Lexicography and Text Research Proceeding of COMPLEX 92.* Budapest: MTA Nyelvtudományi Intézet, pp. 259-274.

PRÓSZÉKY, G. - TIHANYI, L.: (1992) A Fast Morphological Analyzer for Lemmatizing Corpora of Agglutinative Languages. In: *Papers on Computational Lexicography and Text Research Proceeding of COMPLEX 92.* Budapest: MTA Nyelvtudományi Intézet, pp. 275-278.

SALMINEN, A. - TOMPA, F.W.: (1992) PAT Expressions: an algebra for text search. In: *Papers on Computational Lexicography and Text Research Proceeding of COMPLEX 92.* Budapest: MTA Nyelvtudományi Intézet, pp. 309-332.

A Hyperinflation of Lexical Mega-Monsters? *Mega-, Ultra-, Super-,* and *Hyper-* as Intensifying Prefixes: A Corpus-Based Study

ROSWITHA RAAB-FISCHER

Abstract

This study explores the use of the intensifying word-formation elements *mega-*, *ultra-*, *super-* and *hyper-* in modern-day British English. It employs the January 1991 to June 1991 issues of the British newspaper *The Guardian* as a large and up-to-date text corpus. The articles with words containing these intensifying word-formation elements were copied on diskette and read into the text analysis program TACT.[1] The examination of the contexts provides clues to the subject areas and the specific uses of these intensifiers. Keywords can be distinguished that motivate new coinages. Through a corpus study like the present one, and through examining the history of the relevant words, the coining of neologisms can be explained and sometimes even predicted.

1. Introduction

In *A Dictionary of Modern English Usage*, H.W. Fowler condemns the use of the word element *super-* as being barbarous, which, however, did not prevent *super-* from becoming popular. *Mega-*, *ultra-* and *hyper-* have become established in English as well. The "superman" has "superpower", the "megastar" cleans up "megabucks" by making "megadeals", and the "ultrabimbos" are "hyper-elegant" (though not necessarily "hyper-intelligent"). We all seem to be "superlaholics".

The present study explores the uses of words containing the intensifiers *mega-*, *ultra-*, *super-* and *hyper-* in the January 1991 to June 1991 issues of the British newspaper *The Guardian*. It can be observed that these word elements are used in specific contexts, and that they are interchangeable in certain words.

2. The word-formation status of *mega-, ultra-, super-* and *hyper-*

In lexicography and word-formation, *mega-*, *ultra-*, *super-* and *hyper-* are considered to be either prefixes or combining forms. As combining forms, they would enter into compounds; as prefixes, they would form derivatives. Compounds consist of free lexical morphemes, and derivatives have at least one bound grammatical morpheme. However, in English, there exist so-called composite neoclassical lexemes, which contain bound morphemes that carry lexical meaning. Because of their meaning, these lexemes are usually classified as compounds.

According to the *Oxford English Dictionary, mega-* comes from Greek 'large', 'big'. As a unit of measurement, it means 'one million'. *Ultra-*, meaning 'beyond', is derived from the Latin adjectives *ultramarinus, ultramontanus* and *ultramundanus*, and, meaning 'extreme', 'radical', from the French *ultra-révolutionnaire* and *ultra-royaliste*. It has been used as an intensifier

since the 19th century. *Super-* goes back to the Latin preposition *super*, 'above', 'beyond', which itself is a shortening of the Latin *superus*, 'superior'. *Hyper-* is the Greek equivalent of *super-*. The intensifiers *super-* and *hyper-* are figurative extensions of their original meanings. *Super-* has been used as an intensifier since the 16th century, and *hyper-* since the 19th century.

Based on their etymologies, *mega-* and *super-* could be classified as combining forms, and *ultra-* and *hyper-* as prefixes: the former go back to a lexical stem, whereas the latter have their origin in prefixes. From a synchronic point of view, *mega-* and *super-* denote a certain size of a referential object or a degree of intensity of a state or an action. They, like *ultra-* and *hyper-*, are considered to be prefixes because their meaning differs from their original meaning.

3. *The Guardian* on CD-ROM[2]

The Guardian serves as a large corpus which is easy to obtain and to handle. It represents the current use of Standard English. Its articles are divided into the sections *home news, foreign news, city news, features, sport, Euro Supplement, eG Supplement* and *Weekend Guardian*. In an average edition of some 36 pages, *home news* occupy about five, *foreign news* about three to four, *city news* three, *features* six to seven and *sport* four pages. The *home news* mainly consist of articles about Great Britain's domestic and foreign policies, whereas the *foreign news* inform about the rest of the world. The *city news* are on the whole comparable with a business or financial section. The *features* section forms a considerable part of the newspaper because it contains weekly supplements, such as the *Education Guardian, Computer Guardian* and *Media Guardian*. The *Weekend Guardian* is a part of the Saturday edition and comprises fiction, poetry, travel descriptions and portraits of important persons. The *Euro Supplement* deals with Europe and the Common Market, and, finally, the *eG Supplement* contains articles for youngsters and teenagers.

For a linguistic analysis that is based on the restriction of certain words to certain subject areas, the division into sections is only partly useful. If one compares single editions of *The Guardian* on paper with the respective editions on CD-ROM, it becomes clear that not all articles are assigned to the right section. For instance, the articles on the first and last pages are always assigned to *home news*, regardless of their content; or many articles about European topics appear under *foreign news* or even *home news*.

Texts, headlines and bylines can be searched for words. The text frequency of words is given on the whole or in the sections consulted. For the overall frequency of words, one has to browse through all relevant texts. The item searched for is highlighted. If one wants to print out the context of an item or copy it onto diskette, one has to copy the whole article. It is possible to truncate words. One can search for an initial part of a word by adding an asterisk to it. This is not possible with final parts of words.

4. Selection of lexemes

I analysed all lexemes with *mega-, ultra-, super-* and *hyper-* as intensifiers of their base, including hyphenated lexemes and those in quotation marks, but without lexemes written with initial capital letters or word combinations where *mega-, ultra-, super-* and *hyper-* are separate words. Words like *megalith, superannuation, supercilious, supersede* and *hyperbole* are single free morphemes and were thus not taken into consideration. Nor were items considered if the intensifiers retain their original meaning, as is the case in *megabyte, megawatt* ('one million') and *ultramontanist, ultrasuede, superhuman, supernatural* ('above', 'beyond'). Sometimes it is difficult to decide whether a prefix has an intensifying function or not. I included even those items in which the prefix even slightly implied intensification of its base.

The different types of occurence or tokens comprise phonetically identical items written with a hyphen, as two words or together, as well as in single or double quotes. The derivatives

of one base constitute different types, e.g. *hyper-nationalistic, hypernationalism* and *hyper-nationalist*. Inflected forms belong to one lexeme.

5. Analysis

For the study of the different uses of *mega-, ultra-, super-* and *hyper-*, it is essential to take into account their context (including the topic of the text), their word class, the meaning of their base and the frequency distribution of their types and tokens. The division into topic-related subclasses gives insights into the common and different uses of these prefixes.

5.1. *Mega-*

As an intensifier, *mega-* has not been in use for a very long time. According to the third edition of the *Barnhart Dictionary of New English*, it has been used as such since the end of the 1960s.

The Guardian from January 1991 to June 1991 contains 36 types and 60 tokens of *mega-*. The relation between types and tokens shows that most types (29) occur only once, which can be interpreted as an indication of high productivity. Types with relatively many tokens are *megastar* (11), *megabuck* (8) and *megastore* (6). They are institutionalized to a relatively high degree. A high frequency of tokens, the spelling without a hyphen and the occurence in different sections of *The Guardian* speak for institutionalization.

As to the context, four sense groups of *mega-* words can be established: 1. MONEY (e.g. *mega-budgeted, mega-business, mega-fund, mega-million*), 2. SIZE (e.g. *mega-churches, mega-temple, mega-conurbation)*, 3. ENTERTAINMENT (e.g. *megadrums, mega-show, mega-production*) and 4. NON-SPECIFIC (e.g. *mega-trendy, mega-sulk, mega-violence, mega-person*). The different sections and the relation between types and tokens are demonstrated by the following table:[3]

mega-	FEA	CIT	WEE	HOM	SPO	sum
MONEY	5 (8)	5 (7)	2 (2)	3 (5)	1 (5)	13 (27)
SIZE	2 (2)	--	1 (1)	1 (1)	--	4 (4)
ENTERTAINMENT	6 (15)	1 (1)	--	--	--	6 (16)
NON-SPECIFIC	9 (9)	--	4 (4)	--	--	13 (13)
sum	22 (34)	6 (8)	7 (7)	4 (6)	1 (5)	36 (60)

It is not always easy to classify the items. Megamarkets and megastores are predominantly commercial centres, but they are usually large as well. And in the entertaining business, large amounts of money are involved.

Most of the words belong to the first group. They denote good and bad deals (e.g. *mega-loser*). With the exception of *megabuck* and *megastore*, the types occur only once, which means that the *mega-* of the first group is relatively productive (the smaller the discrepancy between the numbers of types and tokens, the higher is the productivity). Words with the same base also belong to the first group: *mega-deal, mega-dealer, mega-fund, mega-fundraising*. In non-specific uses, *mega-* appears to be productive as well. Non-specific uses hint at productivity in general because they are not restricted to particular contexts.

Most of the *mega-* words are nouns. The adjectives belong to the groups MONEY and NON-SPECIFIC. They are written with a hyphen, and each type occurs with one token only, which indicates a low degree of instituationalization and high productivity. The majority of the *mega-* words, not only those of the ENTERTAINMENT group but also those of the other groups, occurs in the *features* section. This can be explained by the fact that the *features* section contains most of the new and colloquial items.

The coining of words with *mega-* can be imagined in the following way. Established words motivate the formation of new items that belong to the same thematic class. *Megastar*, for

instance, prompted the coining of *mega-show* and *mega-hit*. The existence of *mega-buck* and *megastore* gave rise to the coining of *mega-business, mega-budgeted* and others. Finally, the non-specific uses follow. *Mega-chef* and *mega-person*, for instance, are coined analogous to *megastar*. Analogous coining can also be caused by phonological and morphosyntactic features of the relevant form. For instance, the formations analogous to *megastar* are nouns and tend to be monosyllabic (e.g. *mega-chef, mega-hit, mega-drum, mega-show*). Words that lead to the formation of new words are called keywords. The new coinages can develop into keywords themselves (e.g. *mega-show* and *mega-production*).

Furthermore, *super-* words have an impact on the formation of *mega-* words. *Superstar, superstore* and *super-rich*, for example, are the starting point for the coining of *megastar, megastore* and *mega-rich*.

The frequencies of types and tokens, their spelling and their occurence in different sections have to be viewed against the background of the age of the items considered, that is, the diachronic perspective has to be included. How should one otherwise know whether *megastar* influenced the coining of *superstar*, or the other way round? However, usually the synchronic and diachronic perspectives correspond, i.e. words with a high frequency that appear in many sections and are written as one word are very often the oldest ones and therefore keywords.

5.2. *Ultra-*

The etymology of *ultra-* (see above) accounts for its frequent use in political contexts, e.g. *ultra-communist, ultra-leftist, ultra-conservative.*

In *The Guardian*, there are 61 types and 92 tokens altogether. The types do not have more than five tokens, which indicates that the typical *ultra-* word is a nonce-formation. Nearly all *ultra-* words are hyphenated. The political terms hold most of the tokens: *ultra-nationalist* (five), *ultra-right* (three), *ultra-conservative* (three) and *ultra-left* (three). As nouns, they designate persons with certain political beliefs; as adjectives, they attribute political attitudes to persons. Four sense groups can be distinguished: 1. POLITICS (see above for examples), 2. FASHION (e.g. *ultra-chic, ultra-fashionable, ultra-posh, ultra-modern*), 3. ATTITUDES (e.g. *ultra-cautious, ultra-intellectual, ultra-confident, ultra-reasonable*), 4. NON-SPECIFIC (e.g. *ultra-soft, ultra-dry, ultra-large*). Most of the *ultra-* words, mainly nouns, belong to the first group. Most of the adjectives with *ultra-* belong to the other three groups.

ultra-	FEA	FOR	HOM	SPO	WEE	CIT	EUR	sum
POLITICS	14 (18)	6 (8)	4 (5)	--	1 (1)	1(1)	4 (4)	21 (37)
FASHION	7 (7)	--	1 (1)	--	2 (2)	--	--	8 (10)
ATTITUDES	7 (10)	3 (3)	2 (2)	5 (6)	2 (2)	--	--	17 (23)
NON-SPECIFIC	2 (3)	1 (1)	3 (4)	4 (5)	5 (5)	3 (4)	--	15 (22)
sum	30 (38)	10 (12))	10 (12)	9 (11)	10 (10)	4 (5)	4 (4)	61 (92)

The groups POLITICS and ATTITUDES overlap, and so do the groups FASHION, ATTITUDES and NON-SPECIFIC (cf. *ultra-traditional, ultra-egalitarian* and *ultra-romantic, ultra-fine, ultra-clear, ultra-secure*). The first group appears to be the most institutionalized (cf. frequency and relation of types and tokens). The words of the other groups seem to be established only to a very small degree.

The development of the *ultra-* words has started in the area of politics. Their use has extended towards adjectives denoting attitudes and then to the area of fashion. This can be explained by the fact that inner views are mirrored by outer appearance. Finally, the non-specific words are coined analogous to the *ultra-* words of the groups FASHION and ATTITUDES. In contrast to the non-specific coinings, the political *ultra-* words represent a stable, almost closed group. They constitute word groups with the same base (e.g. *ultra-left, ultra-leftist* and *ultra-leftwing*), which can be seen as an indicator of their wide dissemination.

5.3. *Super-*

Most of the types and tokens considered in this study are words with the intensifying prefix *super-*. *The Guardian* contains 132 types and 312 tokens. Apart from that, *super-* is frequently used in proper nouns and as an adjective (both of which are not object of this study). *Superpower* and *supercomputer* show many tokens (20 and 19, respectively). They are followed by *supergun, supergrass* and *superstate* with ten tokens each. The frequency of *supergun* can be attributed to its news value in the Gulf War and in the supergun affair. Therefore it is not an institutionalised word or keyword. Types with many tokens are written as one word, whereas the types with one to three tokens are hyphenated.

The formations of new words with the same base are, among others, prompted by *superhero* (*superheroine*), *supermarket* (*supermar*), *superpower* (*superpowerful, superpowerdom*) and *superstar* (*superstardom*).

super-

FEA	HOM	CIT	SPO	WEE	FOR	EUR	sum
76 (143)	34 (84)	15 (20)	13 (20)	9 (10)	7 (28)	6 (7)	132 (312)

The *super-* words appear in all kinds of contexts, for instance in words relating to means of transport (*super-ferry, super-jumbo*), politics (*super-ministry, super-left*), military (*super-force, superbomb*), economics (*superstore, super-chain*), science and technology (*supercomputer, super-technology*), entertainment (*super-tuned, super-hit*), sports (*super-fit, super-league*) and others. Therefore, it is not useful to set up topic-related sense groups for *super-*. In general, the high frequency of *super-* is an indicator of its non-specific use and of its institutionalization. This does not exclude the possibility that there exist keywords in particular subject areas. It seems that those *super-* words are keywords that motivate the coining of words with the same base (see above). It is true that, with the exception of *superpower*, they do not occur with many tokens. They do, however, occur with at least five tokens, they are written together and, diachronically seen, they have been in common use for some time. In the *Oxford English Dictionary*, the first occurence of *supermarket* dates from 1933, of *superpower* (as a political term) from 1930 and of *superstar* from 1925. However, the door to word-formations with *super-* has already stood wide open for a while. The new space has been conquered and occupied, and the keywords do not seem to stand out (any more?) from the large amount of non-specific *super-* words.

5.4. *Hyper-*

There are 29 types and 110 tokens in the corpus. The two words *hyperinflation* (33 tokens) and *hyperactive* (29 tokens) are responsible for the relatively high number of tokens. *Hyperinflation* is used in connection with the former Soviet Union and South America and is thus a *hyper-* word that has been in use for some time. *Hyperactivity* and *hypermarket* also appear with many tokens (11 and 9, respectively). Words like *hyperbaric, hypertrophic* and *hyperventilation* are not considered here because they represent lexicalized terms in the sciences, very often denoting bodily malfunction. However, these uses of *hyper-* cannot always be clearly separated from its use as an intensifying prefix. This, for instance, is the case in *hyperactive, hyperactivity, hyperkinetic* and *hypersensitive*. Nevertheless, many words can be easily recognized as intensifiers. In general, they occur with only one token (e.g. *hyper-clear, hyper-critical, hyper-elegant, hyper-poor, hyper-rich, hyper-refinement*).

Three sense groups can be distinguished: 1. BODY (e.g. *hyperactive, hyperallergic, hyper-animated*), 2. MONEY (e.g. *hyperinflation, hyperinflationary*) and 3. NON-SPECIFIC (e.g. *hyper-rich, hyper-clear, hyper-arousal*).

The second group appears to be partly motivated by the first group, as in 'to survive hyper-inflation' (metaphorical extension).

hyper-	FEA	HOM	FOR	CIT	WEE	SPO	EUR	sum
BODY	6 (23)	5 (6)	3 (10)	3 (6)	4 (6)	3 (5)	--	9 (56)
MONEY	2 (3)	1 (1)	1 (14)	1 (13)	1 (1)	--	1 (2)	3 (34)
NON-SPECIFIC	14 (16)	--	1 (1)	1 (1)	--	1 (1)	1 (1)	17 (20)
sum	22 (42)	6 (7)	5 (25)	5 (20)	5 (7)	4 (6)	2 (3)	29 (110)

Hyperactive appears to be a keyword. Originally, it referred to physiological activity, and it is now also used to intensify the base word. It influences the formation of *hyper-* words that refer to the body and its functions as well as to non-specific words. One also has to take into consideration that *hyperactive* was clipped to *hyper*. The words *hypermarket*, a loan translation of French *hypermarché*, and *hyperinflation* with *hyperinflationary* stand relatively isolated. The non-specific *hyper-* words are partly analogous coinings to the corresponding *super-* and *ultra-* words (e.g. *super-active/hyperactive, super-rich/ultra-rich/hyper-rich*).

5.5. Comparison

The *features* section contains most of the items, which can be explained by the fact that it is the biggest section in *The Guardian*. Apart from that, the vocabulary in *features* is less topical and more colloquial than in the other sections. In addition, the high numbers of topic-specific tokens can be put down to the high frequency of particular lexemes, e.g. *megabuck* (*sport*, 8 tokens), *megastar* (*features*, 10 tokens), *supercomputer* (*features*, 11 tokens), *superpower* (*foreign news*, 11 tokens) and *hyperinflation* (*city news*, 11 tokens, *foreign news*, 14 tokens). This is not relevant for *ultra-* with less than 6 tokens of each type.

Not much can be said about the specific uses of the prefixes in the different sections. *Ultra-* often occurs in the *foreign news* section, which can be attributed to the fact that most of the *ultra-* words are used in politics. The other prefixes do not occur in this section as often as *ultra-*. The fewest types can be found in the *Euro Supplement* section -- probably because it is a section with a very restricted vocabulary.

Super- has the highest frequency, and it is used in all kinds of contexts. In contrast, *mega-*, *ultra-* and *hyper-* are used only in specific contexts. *Mega-* is often associated with money, *ultra-* with politics and *hyper-* with the body and the senses. Their context-specificity can be related to their origins and to their other meanings. *Mega-* is often used when large sums of money are involved, which can be connected to its original meaning 'one million'. The *ultra-* words are motivated by the French loans (see above). The use of *hyper-* in connection with the body and referring to its functions is motivated by its medical implications. The non-specific uses of the prefixes appear to be influenced by these associations.

In some words, *mega-*, *ultra-*, *super-* and *hyper-* seem to be interchangeable. The following table shows word bases that are combined with more than one of the four prefixes:[4]

BASE	*mega-*	*ultra-*	*super-*	*hyper-*
active	-	-	super-active	hyper(-)active
carrier	mega-carrier	-	super-carrier	-
clear	-	ultra-clear	-	hyper-clear
competitive	-	-	super-competitive	hyper-competitive
confident	-	ultra-confident	super-confident	-
fast	-	ultra-fast	superfast	-
grass	megagrass	-	supergrass	-
hawk	-	ultra-hawk	superhawk	-
hit	mega-hit	-	super(-)hit	-

left	-	ultra-left	super-left	-
lightweight	-	ultra-lightweight	super-lightweight	-
market	megamarket	-	supermarket	hypermarket
power	mega-power	-	super(-)power	-
rich	-	ultra-rich	super(-)rich	hyper-rich
romantic	-	ultra-romantic	-	hyper-romantic
sensitive	-	ultra-sensitive	-	hyper-sensitive
star	megastar	-	superstar	-
store	megastore	-	superstore	-
sweet	-	ultra-sweet	super-sweet	-
trendy	mega-trendy	ultra-trendy	-	-

The corpus analysis suggests that the word class and the meaning of the base are of prime importance for the choice of prefix. Nouns tend to combine with *mega-*, adjectives with *ultra-* and, to a lesser extent, with *hyper-*. *Super-* is added to nouns as well as to adjectives. In connection with finances and business, especially *mega-* nouns form synonyms for *super-* nouns; and in connection with non-specific uses, especially *ultra-* adjectives and, to a lesser extent, *hyper-* adjectives form synonyms for their *super-* correlates. This can be attributed to the fact that *super-* has connotations of importance and uniqueness. Because the use of *super-* has long been established, and its emotional force has consequently been lessened, a need has arisen for other prefixes that are better suited to express significance. These prefixes seem to be *mega-* for nouns and *ultra-* and, to a lesser extent, *hyper-* for adjectives, and it remains to be seen whether they will suffer the same fate as *super-*. By now, their use has been restricted mainly to colloquial language.

The question arises whether the gaps in the above table are systematic or caused by chance. In the context of the present study, seven native speakers were asked to take an elicitation test. They were to mark the uses which they thought to be acceptable. The results differed greatly. One informant, for instance, considered all *mega-* words except *mega-hawk* acceptable, whereas another informant only accepted *mega-rich*. Mainly the younger informants approved of the formations, whereas the older ones rejected them. The test showed that in lexicalised, usually nominal formations, one prefix cannot easily be exchanged for another: formations with *carrier, hawk, market* and *lightweight* were frequently rejected (**ultra-carrier, *hyper-hawk, *mega-lightweight, *ultra-market*). The topic-specific context played a role, e.g. the negative associations of *hyper-* blocked with 'affirmative' bases. Additionally, extralinguistic factors like age, nationality, ethnicity and social background of the informants played an important role.

From a diachronic perspective, the coinage of intensifying *mega-*, *ultra-* and *hyper-* words have been motivated by the corresponding *super-* words, which are similar to them in morphological structure, word class, number of syllables and topic-specificity.

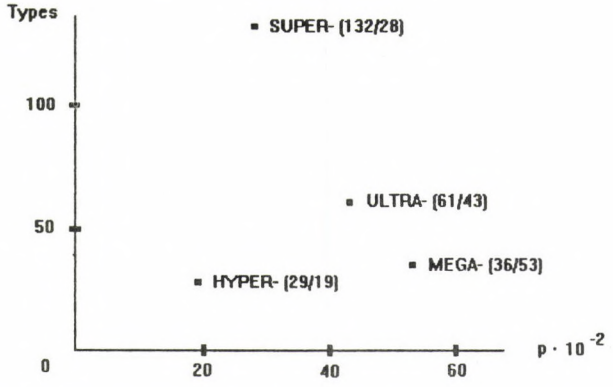
In summary, all intensifying prefixes except *super-* are used in specific contexts, and they carry synonymous meaning in only a few words, which are mainly adjectives.

Which of the prefixes is actually preferred depends on word class, semantic connotations and the stylistic values that are related to extralinguistic factors. *Super-* seems to be 'out', *mega-* and, above all, *ultra-* seem to be 'in'.

5.6. The productivity of *mega-*, *ultra-*, *super-* and *hyper-*

Mega-, *ultra-*, *super-* and *hyper-* are productive, i.e. they enter into new word combinations. Since Aronoff, it has been tried to find a measure of productivity (1976, 35-45). Baayen/Lieber could show convincingly that productivity can be measured by means of the ratio between types and tokens (1991, 801-843). They found out that productivity increases with a growing number of types with one token (hapaxes) and that it decreases with a growing number of types with many tokens (809-820). Productivity can thus be measured by the

quotient of the number of hapaxes and the overall sum of tokens. It describes the probability of the occurrence of new types. The so determined measure of productivity (*p*) is 0.53 (32/60) for *mega-*, 0.43 (40/92) for *ultra-*, 0.28 (86/312) for *super-* and 0.19 (21/110) for *hyper-*. However, productivity is also related to the overall number of types of a word-formation pattern because the number of types is determined by restrictions that limit the application of word-formation rules. The following table shows what is called the "global productivity" (gp), which also includes the number of types



The numbers for the "simple" productivity (*p*) can be qualified with regard to the types. The global productivity (*pg*) of *super-* has to be rated much higher than the global productivity of *mega-*, *ultra-* and *hyper-*. *Hyper-* is the least productive prefix; *ultra-*, *mega-* and, finally, *super-* follow with increasing productivity. If the values of both coordinates differ too much, it becomes difficult to calculate the global productivity. Therefore, Baayen tries to find a measure of global productivity through complicated mathematical considerations. These, however, cannot be verified by lay mathematicians (1993, 12f.).

6. Final remarks

In the English language, *mega-*, *ultra-*, *super-* and *hyper-* are used as intensifying prefixes. In the January to June 1991 issues of *The Guardian*, which was used as a corpus in this study, the preferred topic-related word formation types of *mega-*, *ultra-* , *super-* and *hyper-* were examined. The corpus analysis shows that the institutionalization of the prefixes is determined by different factors. Mainly those words become established that have many tokens (i.e. that are frequent), that are written together, that are nouns and that serve as a base for new word formations. Institutionalization has to be distinguished from relevance. Relevant words, whether institutionalized or not, occur frequently when they are particularly relevant for a language community at a certain time. The so caused high frequency of an item can lead to its institutionalization: *parole* becomes *langue*; innovation in performance further develops competence.

Keywords are usually words that are institutionalized, that are associated with many other words and that motivate new coinings which resemble them with respect to phonological, morphological, morpho-syntactic and semantic features. The new coinages can develop into keywords themselves, and original keywords can become insignificant.

Finally, the question shall be raised which of the examined prefixations will be found in the mental lexicon. Since Chomsky's "Remarks on Nominalizations" and because of the proceeding development of cognitive linguistics, the mental lexicon has become a focus of attention. It is assumed that irregular word-formation processes are stored in the mental lexicon, whereas

regular word-formation processes are stored only partly. Frequently used words are stored as lexical entities in contrast to rule-patterned, infrequent word formations including neologisms, whose single components are stored as well as rules for their linking.

The frequency of words correlates with irregular word formation (see Bybee 1985, 119). The more frequent words are, the more likely it is that they are formed irregularly and that they are stored in the mental lexicon. They form autonomous isles in the pool of word formations. Their associations with other words are less distinct than the associations of less frequent words. At first sight, this seems to be a contradiction of the keyword theory. Keywords are highly frequent words, and they have many distinct associations with other words (their analogous formations). However, one has to indicate the direction of the associations. It is true that keywords are relatively autonomous, but their analogous formations refer to them or lean against them.

According to Aitchison, the mental lexicon provides a lexical tool-kit and a back-up store for the formation of neologisms (1987, 116f., 161). The back-up store consists of word forming elements that are combined to form new words with the aid of the lexical tool-kit. Accordingly, the keywords would be stored in the mental lexicon, and the least institutionlized words would be formed and interpreted by the back-up store and the lexical tool-kit. Applied to *mega-*, *ultra-*, *super-* and *hyper-*, this means that words like ***megamarket***, ***ultra-nationalism***, ***superstar*** and ***hyperactive*** occur as such in the mental lexicon, whereas only the separate components of ***mega-monster***, ***ultra-posh***, ***super-yuppie*** and ***hyper-romantic*** occur in the back-up store. Words that are relevant to a present situation would be stored in the mental lexicon for their time of relevance, e.g. ***supergun*** and ***hyperinflation***. It can be assumed that there exists a kind of transitional area, in which words are sometimes stored and sometimes not, according to a collective or individual need, which is certainly the case with the prefixes examined in this study.

Footnotes

[1] TACT is a freeware program that was developed by John Bradley and Lidio Presutti at the University of Toronto (published 1989).

[2] Chadwick-Healey Ltd., Cambridge, England, is the publisher of *The Guardian* on CD-ROM. Next to that, they have published *The Times, The Sunday Times, Financial Times, The Economist, The Independent* and *The Telegraph*, as well as bibliographies, poetry, drama, indices and a dictionary since 1993.

[3] The tokens appear in brackets after the types. The types of the sums are the overall types without duplicates, i.e. without considering the occurrence of the same types in different sections. For that reason, the sums are sometimes smaller than the sums of the types of the sections. The sections are FEA (features), CIT (city news), WEE (Weekend Guardian), HOM (home news), SPO (sport). The number of items decreases from left to right. The sections EUR (Euro Supplement) and FOR (foreign news) do not contain *mega-* words.

[4] A hyphen in brackets indicates that a lexeme is written together or is hyphenated. -- *Hawk* denotes a person who is pro-war (antonym to *dove*, see *Barnhart Dictionary of New English*).

Literature

Aitchison, J. *Words in the Mind. An Introduction to the Mental Lexicon*. Oxford: Blackwell, 1987.

Aronoff, M. *Word Formation in Generative Grammar*. MIT, 1976.

Baayen, H. "On Frequency, Transparency and Productivity." *Yearbook of Morphology 1992*. Ed. G. E. Booij and J. van Marle. Dordrecht: Kluwer, 1993. 227-254.

Baayen, H., and R. Lieber. "Productivity and English Derivation: A Corpus-Based Study." *Linguistics* 29.5 (1991): 801-843.

The 3rd Barnhart Dictionary of New English. Ed. R. K. Barnhart, S. S. Barnhart and C. L. Barnhart . H. W. Wilson Co, 1990.

Bybee, J. L. *Morphology. A Study of the Relation Between Meaning and Form*. Amsterdam: Benjamins, 1985.

Chomsky, N. "Remarks on Nominalization." *Readings in English Transformational Grammar*. Ed. R. A. Jakobs and P. S. Rosenbaum. Waltham, Mass.: Ginn and Company, 1970: 184-221.

A Dictionary of Modern English Usage. H. W. Fowler. Oxford: Clarendon, 1926.

The Oxford English Dictionary. Ed. J. A. Simpson and E. S. C. Weiner. 2nd ed. Oxford: Clarendon, 1989.

The Debrecen Computational Lexicographical-Terminological Project in Foreign Languages for Special Purposes - the Initial Stage

FERENC ROVNY

Present study tries to give an overview of the project and the preparatory work in connection with it. Teachers at the Foreign Language Centre at Kossuth L. University published a lot of collections of terms and phrases in various fields of science in the past years. Their modernisation in the traditional way is almost impossible and impracticable, so a decision on computerizing the whole process of the collection of data was reached to make regular updating necessitated by constant coinage of new scientific terms easier. More than 10 databases are planned, that has led to the idea that a special centre should be set up for this purpose. Problems about the choice of appropriate hardware / software and the development of a special dedicated lexical software system; and questions about the definition of the type of entry (lexicographical? terminological?) are also discussed.

The teaching of foreign languages for special purposes has been going on for decades at Kossuth Lajos University both for the students of humanities and for the students of sciences in the following languages: English, German, French, Russian and more recently Spanish and Italian. From 1983 onwards, the Foreign Language Centre has also been training translators in the fields of chemistry, mathematics, physics, biology and psychology.

In order to put professional foreign language training on a solid base, the teachers of the department gathered collections of professional terms and phrases in various fields of science, and later they were published in books.

With the rapid development of sciences, most of these books have become outdated by now and there is a strong need to substantially enlarge and modernise them. The traditional way of creating such collections of terms and phrases seems to be inapplicable with the ever growing data and constant coinage of new scientific terms and phrases. That is why we have decided to computerize the whole process of the collection of specialist terms and phrases, and thus the collected data can regularly be modified, supplemented, corrected and updated.

Main objectives of the project

1. The establishment of a computerized terminological database, accessible for a wide range of students, teachers and specialists. This will include 12 different databases, each containing up to 25,000 - 30,000 entries with a total of 360,000 entries, which may require up to 16 Gigabytes when enlarged by audio and visual information.
2. The regular enlargement of the database.
3. Providing computer services on the basis of the database(network, floppies, CD, etc.)
4. Modernization of the present collections of specialist terms and phrases.
5. Editing various publications on the basis of the database and offering them for publication.

1.1. **Definition and justification of the major objectives of the project**

1.1.1.

The establishment of a computerized terminological database in biology, chemistry, physics, computer science, mathematics, psychology and geography, accessible for a wide range of students, teachers, and specialists in the following languages: English, German, French and Spanish in accordance with the development of particular fields of science.

1.1.2

According to an agreement between universities it is the task of the Foreign Language Centre at Kossuth L. University to make bilingual collections of scientific terms and phrases (specialist vocabularies) in Hungary. **There are no bilingual scientific specialist vocabularies** (apart from our former publications) as we have been informed by OMIKK (National Technical Information Centre and Library) and the Library of Kossuth University (the 2nd National Library). The exception is the field of computer science, but computer dictionaries published are at a very mixed level and they do not reflect the whole field, a kind of integration, the making of a new dictionary based on up-to-date database is indispensable even in this field. Information obtained from OMIKK proves that **there is no computerized terminological database in Hungary**. That is why we attach great importance to the fact that users should have a database that meets the requirements of our age, stored on data

media, flexible, amenable to upgrading, enlargable and up-to-date. Relying on the database a special (computerized) service centre is to be set up with various possibilities of use.

I.2. Institutional background

1.2.1.
There are three main areas covered by the Foreign Language Centre as follows (two educational, one combining research work and publications): 1.2.1.1.
Teaching foreign languages - both general and special purposes - for all the students of the whole university (with the exception of students specialised in foreign languages).

Table 1.

The total number of (day-course) students at Kossuth Lajos University.

	1990	1991	1992
Number of students	3151	3420	3045

Table 2.

The number of students taught by the Foreign Language Centre:

	1990	1991	1992
English	550	430	470
German	275	224	314
French	80	80	107
Spanish	103	72	70
Italian	75	38	35
Russian	208	138	120
Total	1291	892	1116

The difference in numbers arises from two facts: 1. students have to study two foreign languages only for 3 (2+1) years, 2. some of them already have a "State Language Examination Certificate at Intermediate Level", that means they do not have to attend language lessons according to regulations.
It is a different problem though that by having a **general** language examination they have been exempt from studying **specialist** foreign language. Seeing the illogical nature of this regulation, we have proposed they should have one term in specialist foreign language and our proposal has just been accepted by the University Council.

1.2.1.2.
Furthermore, the training of professional translators in English has been going on for 10 years in the following fields: mathematics, physics, biology, chemistry and psychology. Training in German has just started. Besides day-course students, specialists also take part in this training as a form of postgraduate course.

Table 3.

Number of students in translator training:

	Course	1990	1991	1992
Mathematics	2-5	43	49	61
Physics	2-5	16	10	27
Chemistry	2-5	43	45	57
Biology	2-5	32	28	38
Psychology	2-5	-	-	-
Postgraduates total:			14 (1990-92)	

1.2.1.3.
In order to make educational work more efficient, considerable research work was done concerning terminology in various fields, then various educational aids, supplementary materials, collections of terms and phrases were made and a number of them were published, too.

Table List 4.

Bilingual collections of terms and phrases and other books earlier published:

1. I. Nyirkos-J. Mojzes: Collection of English and Hungarian Terms and Texts in Chemistry. (1974)
2. A. Ménes: Collection of Russian and Hungarian Terms and Texts in Mathematics. (1975)
3. A. Ménes: Collection of English and Hungarian Terms and Texts in Mathematics. (1977)
4. I. Nyirkos-J. Mojzes: Collection of Russian and Hungarian Terms and Texts in Chemistry. (N.d.)
5. L. Pősze-L. Molnár Collection of English and Hungarian Terms and Texts in Physics. (N.d.)
6. L. Pősze-J. Bacsó: Collection of Russian and Hungarian Terms and Texts in Physics. (N.d.)
7. I. Nyirkos-I. Hatvani: Collection of English and Hungarian Terms and Texts in Biology. (1980)
8. J. K. Buzáky-I. Korondán: Collection of German and English Terms and Texts in Chemistry. (1981)
9. L. Kornya: Collection of German and Hungarian Terms and Texts in Mathematics. (N.d.)
10. L. Pősze-J.K.Buzáky-L.Kornya: German for Beginners (N.d.)
11. L. Kornya: Civilization and Culture of German - speaking countries: Texts and Exercises (1981)

Besides the above-listed publications a huge amount of other source materials and other collections of terms and phrases are at our disposal.

1.2.2. Staff available
At present, altogether 30 teachers work at the Department of Foreign Languages as follows:

The distribution of teachers (1992):
English: 19, German: 6, French: 2, Russian: 2, Spanish: 1, Total: 30.

Both teachers taking part in translation training leading to a degree and teaching the usual curriculum for regular students, more or less specialise in a particular field. (Naturally, it is especially true in the case of English and German.) Staff members have compiled several readers and glossaries (see Table 4.). Most of them have been teaching translation and reading for special purposes and working in the field of terminology. It is, however, necessary to brush up their knowledge of computers and to recruit staff for administrative purposes (mainly data input).

1.3.
Existing technical equipment
Existing technical equipment relevant from the point of view of this project is fairly poor now, so significant improvement of present computational background is also highly important (buying new, higher performance computers and peripherals).

2.1. Detailed description of the project

2.1.1.
According to the new aspects of our objectives, preparatory studies are necessary:
2.1.1.1.
The participation of all the teachers taking part in courses on operating computers is also indispensable.
2.1.1.2.
Study trips prior to the beginning of work to gain information are also of vital importance.
Study trips are planned to the United Kingdom, France and Germany to study existing computerised lexicographical-terminological databases: Longman, Oxford, etc., and centres in France and Germany.
We also plan to invite foreign experts for an exchange of views and experiences and to ask for their advice on making computerized dictionaries. Acquiring new books, journals, CD-s etc. for the library is also necessary.
As we have formerly mentioned, the installation of suitable computer equipment with the necessary peripherals and software is a key problem of the success of the project.

Our concrete aims are as follows:
First we want to **computerize existing data**, that is the collections of specialist terms and phrases already published.
The second step would be the **enlargement and updating** of the vocabularies of particular fields with the help of students. We plan that "regular" students – besides translating 10 standard pages of specialist text (compulsory before the university language exam), will collect specialist terms and phrases in the text in a **special entry form** (instead of the former "school-like vocabulary") according to points of view defined by us. Students training to be professional translators also contribute to the project – 50 entries per terms plus 200 entries accompanying their final translation. Then the teacher checking their translation checks their

entry forms as well or if a student uses a floppy disc to input data, checks data on the disc. If deemed necessary, specialists – mainly those reading a particular field at our university – also check the correctness of data.
In the third phase – if the input of the database has already reached a considerable extent – will our aim be realized: **providing different services** based on the database.
These are as follows:

a) By means of the computer network of Debrecen Universitas (partly installed at the moment and fully operational by the time of the realisation of the project), which will make them widely accessible.
b) By means of data media: floppy discs, CD, perhaps audiocassettes.
c) By the publication of dictionaries of specialist terms and phrases:

2.2. Funding of the Project

The implementation (or at least the starting) of the project is mainly funded by our successful application in the framework of the Fund for Catching up with European Higher Education Program for the Development of Foreign Language Teaching - 3. About half of the financial support is planned to be spent on procurement (computers, peripherals, software, books, journals, CD-s etc.) and the other half mainly an "human resources"(technical assistance of foreign and Hungarian experts, software development, data input, operational costs etc.).

2.2.1.
The realization of the project would be impossible without the necessary technical equipment (hardware, software) which in turn are defined by the volume of the project.

2.2.1.1.
Requirement for memory and storage capacity of the computers
During the creation of the computerized terminological database and the various collections of words and phrases(specialist vocabularies) we have to take into consideration that textual information only is not sufficient or adequate for the definition of the terms (e.g.: biology - especially botany and zoology; geography, chemistry - models, etc.), so it is necessary to have both visual and audio information. Being a foreign language database, the **pronunciation** of words and phrases should also be given (at least, in some of the languages).
When processing visual information (as far as a process is to be shown, so the task is the digitizing of motion /video/ pictures) we want to rely on the Multimedia Centre (and its equipment) established as a result of a former successful application submitted by our university for the Fund for Catching up with European Higher Education Program for the Development of Foreign Language Teaching.
On defining the capacity of the computers to be used we must not forget that four (five) languages and seven fields of science are included and the database in its planned final form can reach 30,000 lexical units (entries) in each of several fields, that is only the textual part in foreign language only and audio-visual information is to be associated with it, too.

A.) **The volume of the project**
The project involves four languages and seven special fields; however as we have a full-scale specialist vocabulary only in the English language, actually twelve terminological databases are planned at the beginning.

A maximum of 30,000 foreign language headwords are planned for one database (altogether 360,000 headwords) and we have to add the following:
- expressions formed by the headword (2-4),
- foreign language model sentences (1-2),
- Hungarian shades of meanings,
- detailed explanation of the term in a lot of entries,
- the pronunciation of the word/expression (in English the phonetic transcription as well),
- and, if it is possible or needed, the visual information, too.

B.) **Formation of an entry**

Data	Storage capacity requirement	
I. TEXTUAL PART		
Headwords, phrases, example sentences and their Hungarian equivalents, occasionally the explanation of the headword	approx.	4 kilobytes
II. COMPRESSED VOICE (PRONUNCIATION)	approx.	13 kilobytes
III. COMPRESSED PICTURE	approx.	20 kilobytes
Total :	approx.	37 kilobytes

C.) **Storage capacity requirement** of the complete database of one special field, one language:

maximum 30,000 entries	1,100,000 kilobytes
Memory capacity requirement of the whole database (12 special fields)	13,320,000 kilobytes
Memory capacity requirement for operating the database (20"%) 2.664,000 kilobytes	
Total memory capacity required for the whole database:	15,984,000 kilobytes **(approx. 16 Gigabytes)**

During the processing approximately ten times this amount of data should be run.

2.2.1.2. **Computers, peripherals and software programs to be acquired:**
A) A UNIX workstation
1. Workstation, expandable (to more processors) with UNIX operation system; model with fast, high resolution graphics, 64 MB RAM, 1 GB inner HDD; 20" colour monitor, etc. (performance: min. 101 MIPS)
It will be supplemented by HDD with a storage capacity of approximately 5 GB at the beginning, a CD-ROM drive and a DAT unit to store accumulated data.
B) 3 or 4 PC-s are to be used as terminals (Two of them relatively high performance, capable for the development of a PC based end-product – a computerized dictionary program.
C) Suitable peripherals are also needed for data input: scanner, laser printer and other (multimedia) equipment such as a soundcard.
D) The procurement of different software programs are also planned, but only as supplementary. According to a newer concept – owing to developments after the application in 1993 and changes in the trends of computer market – we want to and have to realize a **multiplatform development** when making the software for the computerized dictionary.

The amount of data is so vast that it can only be processed and managed safely by a UNIX workstation – which is also capable of providing network services –, so there is no change in that. But the majority of users, as shown by market tendencies, still use PC, due to the constant development of hardware (higher performance and falling prices) and the even richer choice of programs etc. So the computerized lexical-terminological database should be run on PC as well (we have think of CD, first of all).

3. A short sketch of the computer system to be implemented for the lexical-terminological database

A) The information (text, image, voice data) gets into the memory storage (e.g. HDD) of the computers (both platform: UNIX–SUN and WINDOWS–PC) by means of peripherals (scanner, keyboard, microphone–soundcard etc.) and their special software programs.

B) By the way of the **bi-platform multimedia hypertext system** to be developed, the lexical database will be available by complex retrieval techniques in computer network, compact discs (perhaps floppy discs) and publications.

We have decided upon the development of our own software system for the lexical database for manifold reasons:

a) by sheer reasoning,

b) by the lack of limitless financial support to have a try at all the (commercial, "business") database management systems with various subsystems claimed to be "perfect" for the task by their vendors. "You (or rather we) only have to adapt the system a little bit; so buy the main system and (two or three) subsystems and the adaptation and the development of a special module will only take about half a year (or more) and thousands of USD extra."

c) and last but not least, by studying relevant reference literature:

"The functionality of *general purpose database management systems* e.g. relational ones - is too limited for lexical databases because they are not tuned to the task at hand; in particular, they do not provide for a formalism which is suited to describe *linguistic knowledge*." And: "A dedicated system supports the construction, use and maintenance of lexical databases much more directly than a general purpose database management system in conjunction with a conventional programming language interface." (DOMENIG 1988, p. 154.)

"During the experiments with the lexical database, program developers came to the conclusion that text databases essentially differ from other – e.g. business or statistical – databases." And: "Initial results (with the GOEDEL) are promising, this new approach seems to match the requirements for the management of text-oriented databases much better than traditional relational database management systems." (PAJZS, 1990, P. 18. – speaking about the computerization of the New OED and the GOEDEL programming language.)

4) The Organizational Structure of the Project

The work of the Project Manager of the multi-lingual lexicographical database is supported by teachers responsible for a given language (English, German, French, Spanish). Besides, there are separate persons responsible for collecting terms and phrases in a particular field of science in a given language.

The creation of lexical-terminological databases are planned in the following languages and fields of science:

English: biology, chemistry, computer science, geography, mathematics, physics, psychology; German: chemistry, mathematics; Spanish: biology - botany; French: mathematics.

5) **The present state**

a) The development of the type of entry to be used in the lexical-terminological database is now under way and of course, there are a lot of problems and differing views – amongst us and well-known lexicographers as well. For example, **which pronunciation norm** should be used (in the case of English)? "For EFL purposes, in particular, there is a good case for reflecting both major pronunciation norms in a bilingual dictionary, with double pronunciation entries wherever RP and General American diverge." (WELLS, 1985, p. 46.)
Another, more important problem: **What types of information** should be included in an entry? To what extent should the database be "**lexicographical**" or "**terminological**" in nature? If (rather) terminological, how far should the recommendation referred to in AL-KASIMI, 1983, p. 156, be followed?
Our present entry (see Appendix) seems to represent a mixed type. Is such an approach permissible or reproachable by standards of the art?

b) If the entry is accepted by us, then we would like to have it scrutinized, checked and critisized by well-known linguists and lexicographers.

c) After accepting the final version of the entry (modified by criticisms), the development of a dedicated software system can be initiated.

The (even partial) implementation of the project can give a significant impetus to the raising of the knowledge of specialist foreign language amongst experts (students, lecturers, scientists and experts).

Assistance and co-operation of linguists and lexicographers throughout Europe to successfully carry out our project would be highly appreciated.

References:

1. Marc DOMENIG: The Word Manager in COLING'88, Budapest; Proceedings of the 12th International Conference an Computational Linguistics, p. 154.
2. Julia PAJZS: Computer and Lexicography (in Hungarian), 1990, Linguistica, Series A, Studia Et Dissertationes, 4. Hungarian Academy of Sciences, Research Institute for Linguistics, p. 18.
3. J. C. WELLS: English Pronunciation and its Dictionary Representation, in Dictionaries, Lexicography and Language Learning. Edited by Robert Ilson 1985 Pergamon Press and The British Council, p. 46.
4. A. M. AL-KASIMI: The Interlingual / Translation Dictionary, in Lexicography: Principles and Practice, Edited by R. R. K. Hartmann, Academic Press, Inc., 1983, p. 156.

APPENDIX: **Present Version of the Entry Form**
(1994. 05. 14.)

1. headword :
2. equivalent(s) in foreign language :(⇨)
3. part of speech branching and label(s) :
4. pronunciation(s) :
5. grammatical information :
6. abbreviation in foreign language :
7. abbreviation in Hungarian :
8. a) meaning(s) in Hungarian :(⇨,✱)
   b) functional shades of style :
9. special field; part(s) of field :(⇨)
10. explanation of the headword (term or expression) :(⇨)
11. model sentence(s) in foreign language :
12. Hungarian translation of the model sentence(s) :
13. a) derivative(s) of the headword :
    b) meaning(s) in Hungarian and equivalent(s) :(✱)
14. a) phrase(s) with the headword :
    b) meanings in Hungarian and equivalent(s) :
    c) style :
15. further cross-references
    a) other derivatives of the headword :(⇨)
    b) other phrases with the headword :(⇨)
    c) another headword :(⇨)
16. frequency :
17. voice :
18. image/illustration :
19. remark :

Information on (first) editing:

20. source(s) (title, page, author, date) :
21. collector(s) name :
    ... year, field :
    date of birth :
    address :
    telephone number :
22. linguist's name :
    date(s) of checking :
    remark(s) :
23. name(s) of specialist(s) :
    date(s) of checking :
    remark(s) :
24. name(s) of operator(s) :
    address :
    telephone number :
25. date(s) of updating(s) :

Information on (secondary, tertiary, ... etc.) editing(s):

20.
21.
22.
23.
24.
25.
26. type of modification :

Key to the signs used:

⇨ cross-references,

✱ the given meaning is designated to be a headword in a quasi Hungarian - Foreign Language Dictionary (actually referring back to foreign language headword).

How a Morphological Lexicon for the Italian Language Can Deal with Enclitic Pronominalisation

JACQUELINE VISCONTI

Abstract

This paper presents the results of a study of enclitic pronominalisation, carried out as part of a project concerning speech synthesis and recognition to improve the efficiency of a morphosyntactic parser for the Italian language. The lexical entries of the system are morphemes (prefixes, roots and suffixes), organised into sub-lists and ordered inside a word according to a system of pointers. The verbal inflectional suffixes of the infinitive, the participle, the gerund and the imperative all point to the clitic sub-list. In order to avoid the problems of overgeneration and of production of agrammatical verb-clitic strings we elaborated a constraining device which models the verbal features governing enclitic selection. After implementation of such a model, the morphological lexicon successfully relates indefinite and imperative verbal forms with their respective enclitic pronouns, providing the correct input for the syntactic parser.

0. Introduction

The clitic microsystem is one of the most peculiar and complex areas of the Italian language[1]. Clitics (from Greek *klìnein*, "to bend, to lean") are unstressed pronouns, which cannot occur independently in speech but "lean on" other linguistic forms (usually verbs) to form one complex constituent. These particles can occur either in the preverbal or postverbal position, depending on the verbal morphology. The former case (proclisis) holds for finite verbs (indicative, subjunctive and conditional forms):

la vedo	'I **her** see'
che **la** veda	'that I should **her** see'
la vedrei	'I would **her** see'

the latter (enclisis), which constitutes the specific object of our study, applies to non finite verbs (infinitive, gerund, participles and affirmative imperatives):

veder**la**	'to see **her**'
vedendo**la**	'seeing **her**'

[1]For a general descriptive account see: Battaglia and Pernicone (1954); Berretta (1985a); Brunet (1978-1985); Busch (1985); Calabrese (1985); Cordin and Calabrese (1988); Cordin (1988a); Lepschy and Lepschy (1977); Scrianni (1988), Seuren (1974); Simone (1983). For the categorial status of clitics in the latest version of the G. B. theory (Chomsky 1981; 1992) we follow Belletti (1993); Rizzi (1993); see also *A Bibliography of Clitics: 1892-1991*, Chae, Chair, Nevis, Wanner and Zwicky (eds.), Linguistic Society of America, University of California (Preliminary version, July 1991).

vista**la** '(having) seen **her**'

prendi**la**! 'take **her**!'

There are eleven phonologically (and graphically) distinct clitics, each of which corresponds to several syntactic functions:

ci ([ʧi]), **gli** ([λi]), **la** ([la]), **le** ([le]), **li** ([li]), **lo** ([lo]), **mi** ([mi]), **ne** ([ne]), **si** ([si]), **ti** ([ti]), **vi** ([vi])

Nowhere else in the language are Case distinctions overtly realized through Case inflected forms, which manifest a 4 Case paradigm of Accusative (**lo, la, li, le**), Dative (**gli, le**), Genitive (**ne**) and Locative (**ci, vi**). Moreover, the clitic system morphologically manifests gender distinctions (**lo, la**), person and number features (**mi, ti, lo, ci, vi**) and the distinction between pronouns (**lo**) and anaphors (**si**)[2]. More specifically, we have 5 clitic lists:

	I	II	III	I	II	III
1. Accusative	**mi**	**ti**	**lo,la**	**ci**	**vi**	**li,le**
2. Dative	**mi**	**ti**	**gli,le**	**ci**	**vi**	**loro**[3]
3. Reflexive	**mi**	**ti**	**si**	**ci**	**vi**	**si**
4. Locative	**ci,vi**					
5. Genitive	**ne**					

As can be seen, the relationship between form and function is marked by a strong overlapping of functions and distinctions. A grammatical verb-clitic string results from a satisfactory match between the morphosyntactic and semantic characteristics of the verb and the syntactic and semantic function of the clitic. Intransitive verbs cannot, therefore, subcategorize direct object (accusative) clitics:

*naufragar**lo** 'to shipwreck **it**'

verbs which do not subcategorize a [PP*a* [DP...]][4] cannot select indirect object (dative) clitics:

*illuder**gli** 'to illude **to him**'

verbs which have no locative value cannot select locatives **ci/vi**:

*ricordar**ci** 'to remember **there**'

Native speakers master unproblematically this set of interactions. But how can a morphological lexicon deal with it? How can it recognise as a "non-word" a string like *naufragar**lo** and avoid producing it? To illustrate the problem we shall briefly describe the morphological parser for the Italian language implemented in the Olivetti "Speech and Language" laboratory (Cericola, Danieli, Mollo, Voltolini, 1989).

1. The Olivetti "Speech and Language" morpho-syntactic analyser

This system, elaborated to provide linguistic information for the tasks of speech recognition and text to speech synthesis, integrates both a lexical data-base handling about 2,000,000 forms and a probabilistic syntactic parser. The input can be either in the form of words or sequences of words, written in their graphemic form or in their CPA[5] phonetic transcription. The first step of the analysis, performed by the morphological lexicon, expands upon any

---

[2]On the issue of Italian **si** see: Castelfranchi and Parisi (1976); Cinque (1976); Cordin (1988a); Leone (1979); Lepschy (1974; 1976; 1989); Lo Cascio (1974); Napoli (1976); Parisi (1976).

[3]The dative plural form is expressed by **loro**, which does not share all properties of the clitic forms and does not therefore fall within the scope of this work.

[4]For the internal structure of nominal phrases we follow the DP hypothesis (Abney, 1987), according to which the fonctional category D (Determiner) selects a lexical complement NP with a nominal head:
[DP[D'[D···][NP[AdjP···][N'···]]]].

[5]Computer Phonetic Alphabet, developed for all European languages under the ESPRIT project 860.

lexical ambiguity of the word, adding to each hypothesis all the information available in the lexicon. The result is a lattice in which every word is replaced by one or more quadruplets: {graphemic form, phonetic transcription, grammatical tag, lemma}.
The second step, performed by the syntactic component, takes these lattices as input and yields: the most likely sequence of quadruplets in the lattice; the most likely syntactic interpretation for the chosen sequence and a copy of the chosen text containing prosodic marks. As a result, all the possible syntactic interpretations are ready for further processing.
The morphological component (Delogu, 1989) can analyse and generate a lexicon of about 60 000 words, integrating the graphemic and phonetic transcription of the word with syntactic information and hypotheses of grammatical categorization. The lexical entries of the system are morphemes (Scalise, 1983), which are subdivided into lists of roots and affixes (nominal, verbal, adjectival and adverbial), to improve productivity and computational economy. The combinatory possibilities between lists of morphemes are ruled by a system of pointers, allowing acceptable forms only, in both flexional and derivational morphology. The syntactic analyser uses a bottom-up chart-parsing algorithm with a probabilistic augmented context-free grammar for Italian, consisting of 486 rules, 127 conditions expressing syntactic and lexical constraints and a list of 378 idiomatic expressions. At the end of the analysis a decisional procedure assigns the most plausible interpretation to ambiguous strings, according to their weight in probabilistic terms. The information provided by the morphological lexicon increments the efficiency of the parser in this task. This information should therefore be as complete as possible, if the performance of the whole system is to be improved. To clarify this point, let us return to the lexical component and focus on the issue of the interactions between verbs and clitics.

1.1. Verbal affixes and the clitic list

The morphological lexicon contains 8800 verbal roots and a system of pointers linking roots, thematic vowels, and morphemes of tense and person. The inflexional suffixes of the infinitive (*re*), the gerund (*ndo*), the participle (*to, ta, ti, te*) and the imperative (*i, a, iamo, te*) point towards the clitic sub-list, which is elaborated in order to account for the personal concordance between verb and pronoun:

e.g.:

- infinitive: ved-e-*r*-**ti** 'to see **you**'
- gerund: dorm-e-*ndo*-**vi** 'sleeping **there**'
- participle: am-a-*ta*-**la**. '(having) loved **her**'
- imperative: vend-*i*-**lo** 'sell **it**'

No restriction operates so far on the set of interactions between verbal forms and clitics. The analyser allows the cliticisation of any element of the pronominal microsystem to any member of the verbal class. To solve this problem we designed a device which would control the mapping between morphosyntactic and semantic verbal features and syntactic and semantic clitic functions, thus orienting the analyser to perform correct enclisis. From the point of view of text to speech synthesis, this model contains instructions which uniquely produce acceptable verb-clitic combinations, preventing the cliticisation of: accusative case clitics to intransitive verbs; dative clitics to verbs which do not subcategorize a [PP*a* [DP...]]; locatives **ci/vi** to

verbs which have no locative value; *etc.* From the point of view of speech recognition, the model will allow the analyser to correctly segment the verb-clitic strings, assigning to the clitic an interpretation compatible with the valencies of the subcategorizing verb. This device, the structure and performance of which we shall describe in the following section, models the verbal features governing the selection of enclitic pronouns[6].

2. Elaboration of the model

Given the structural complexity of the clitic system, where every item is related to several functions, the design of such a model demands a careful study of the behaviour of unstressed pronouns in contexts of enclisis, to highlight the interactions between the functional values of the clitics and the morphosyntactic and semantic features of the verbs (Lo Cascio, 1970).

The model, which synthesizes the results of this study in a structure accessible to the machine, departs from three basic assumptions:

- there is a finite number of relevant features which identify a finite number of verbal categories;
- every feature assigns to the corresponding category a finite series of enclitic forms;
- the intersection of the features specifies, for every verbal category, the set of possible choices within the clitic system, excluding forms not allowed by the verbs' valencies.

The impact of these assumptions was evaluated through the analysis of a vast *corpus* of sentences containing verb-enclitic sequences, in order to detect the features relevant to the selection of a specific clitic.

The model is based on three kind of verbal features:

morphosyntactic , such as:
INTRANSITIVE
INACCUSATIVE
COPULATIVE
CLITIC CLIMBING
TRANSITIVE
REFLEXIVE-PRONOMINAL

structural, such as:
SUBCATEGORIZING[PP*a*[DP...]][+ANIM]
SUBCATEGORIZING[PP*a*[DP...]][-ANIM]
SUBCATEGORIZING[PP*(di, da)*[DP...]]
SUBCATEGORIZING[PP*(in, a, per)*[DP...]][LOCAT]

semantic, such as:
"SPACE-LOCATIVE"
"MOTION FROM".

Every feature, as we will see in the following sections, assigns to the corresponding verbal category a finite series of enclitic pronouns, according to the syntactic and semantic functions of the particles.

2.1. Morphosyntactic features

The feature INTRANSITIVE excludes for such verbs the combination with the accusative clitic list **{mi, ti, la, lo, ci, vi le, li, }**, with the reflexive list **{mi, ti, si, ci, vi, si}**, which is distinctive for the REFLEXIVE-PRONOMINAL category, and with partitive, or quantitative, **{ne}** as well as with prepositional **{ne}** for a PP depending on a subcategorized DP:

---

[6]For a computational account of French cliticization within a French Interactive Parsing System, see Laezlinger and Wehrli (1991).

***alluderlo**	**'to allude it'**
***appartenendola**	**'belonging her'**
***tramontarsi**	**'to set itself'** (like the sun)
***esitandone** una (delle ragazze)	'hesitating **of them** one (of the girls)'
***rinunciatine** i genitori (dello sposo)	'(having) renounced **of him** the parents'

The constraint on the particle **ne** does not hold for verbs marked by the combinatory of the features INTRANSITIVE-AUX. "TO BE" (such as *arrivare*, 'to come, to arrive', or *appassire*, 'to wither') and for ERGATIVE verbs (such as *affondare*, 'to sink', *aumentare*, 'to augment', *cambiare*, 'to change', *migliorare*, 'to improve'), all belonging to the more general category of the INACCUSATIVE (Burzio, 1986; Salvi, 1988). These verbs can select both partitive and prepositional **{ne}**:

appassit**one** uno (dei fiori)	'(having) withered **of them** one (of the flowers)'
arrivand**one** la metà (dei ragazzi)	'arriving **of them** half (of the boys)'
essend**one** giunti gli autori (dell'opera)	'having **of it** arrived the authors (of the work)'
sprofondat**ane** la prua (della nave)	'(having) sunk **of it** the prow (of the ship)'

Two more sub-sets of the INTRANSITIVE require a non-standard treatment: the COPULATIVE and the CLITIC CLIMBING. The former, containing verbs like *divenire, diventare*, 'to become', *essere*, 'to be', *parere*, 'to appear', *restare*,'to stay', *riuscire, sembrare*, 'to seem', gives rise to predicative constructions, such as:

sembrare [intelligenti]	'to seem [intelligent]'
diventare [dottore]	'to become [a doctor]'
essendo [sicuro [della scelta]]	'being [sure [of the choice]]'
rimanere [fedele [a lei]]	'to remain [faithful [to her]]'

The predicative complement subcategorized by these verbs, or a PP depending on it, can be pronominalized by a clitic form:

sembrare [intelligenti]⇒sembrar**lo**	'to seem **it**'
diventare [dottore]⇒diventar**lo**	'to become **it**'
essendo [sicuro [della scelta]]⇒essend**one** sicuro	'being **of it** sure'
rimanere [fedele [a lei]]⇒rimaner**le** fedele	'to remain **to her** faithful'

Hence, the restrictions holding for the INTRANSITIVE do not hold for the COPULATIVE. These have been grouped in a closed list and classified as a sub-class of the NON-PRONOMINAL INTRANSITIVE which is allowed access to the predicative **{lo}**, to the prepositional **{ne}**and to the dative clitic list **{mi, ti, gli, le, ci, vi}**.

The CLITIC CLIMBING category, on the other hand, comprises MODAL verbs (*volere*, 'to want', *potere*, 'to be able to', *dovere*, 'to have to', *sapere*, 'to know'), ASPECTUAL verbs (*cominciare*, 'to begin', *finire*, 'to finish', *continuare*, 'to continue') and MOTION verbs (*venire*, 'to come', *andare*, 'to go', *tornare*, 'to return'). These forms manifest an optional encliticisation either to the related infinitive or to the main verb itself, to which the clitic "climbs" (Rizzi, 1976; Berretta, 1985b):

potendo raggiungerl**a** = potend**ola** raggiungere	'being able to reach **her**'
dovendo lavar**si** = dovend**osi** lavare	'having to wash **himself**'
per andare a trovar**la** = per andar**la** a trovare	'in order to go to see **her**'
venite a salutar**mi** = venite**mi** a salutare	'come to greet **me**'

The whole category, although some of these verbs are INTRANSITIVE, has thus been set in a closed list together with the AUXILIARIES (*essere* 'to be' and *avere* 'to have') and assigned the complete clitic list **{mi, ti, la, lo, le, li, gli, ci, vi, si, ne}**[7].

The feature TRANSITIVE links the verbs with the accusative clitic set **{mi, ti, la, lo, le, li, ci, vi}** and partitive and prepositional **{ne}**:

salutandoli	'greeting **them**'
mangiarne due (di mele)	'to eat **of them** two (apples)'
conoscendone la sorella (di Luca)	'knowing **of him** the sister (of Luca)'

Finally, the REFLEXIVE-PRONOMINAL category comprises "intrinsic reflexive", or "pronominal", verbs (such as *vergognarsi*, 'to be ashamed' or *pentirsi*, 'to repent') and "proper reflexive" or "reciprocal" forms (like *lavarsi* , 'to wash himself' or *guardarsi*, 'to look at each other'), the only difference being that is impossibile for the latter to assume INACCUSATIVE value. The feature assigns to the whole class the reflexive list **{mi, ti, si, ci, vi, si}**:

laureandosi	'graduating'

and furthermore excludes the accusative list:

*vergognandoselo	'being ashamed **it**'

Once defined, the set of morphosyntactic features was applied to the 8800 verbal forms contained in the morphologicon lexicon, which were classified in the main categories of INTRANSITIVE (sub-classes: INACCUSATIVE, COPULATIVE, CLITIC CLIMBING), TRANSIT' 'E and REFLEXIVE-PRONOMINAL. The whole verbal class was then further investigated, in order to achieve a structural analysis of the subcategorizing properties of each verb (Elia, Martinelli and D'Agostino, 1981).

2.2. Structural features

This set of features refers to the nuclear structure of the sentence, *i. e.* to the complex of nuclear elements, or arguments, subcategorized by the verb (Salvi, 1988). The first targets of the analysis were verbs that take an argumental [PP*a*[DP...]], like *dare*, 'to give'. The pronominalisation depends in this case on the opposition [+ANIM] *vs* [-ANIM] attributed to the referent of the subcategorized PP. The feature SUBCATEGORIZING[PP*a*[DP...]][+ANIM] assigns to the verbs the dative clitic list **{mi, ti, gli, le, ci, vi}**:

appartenergli (a Gianni) [+ANIM]	'to belong **to him** (to Gianni)'
avvicinarglisi (al bambino) [+ANIM]	'to get closer **to him** (to the child)'
riferirle (alla maestra) [+ANIM]	'to report **to her** (to the teacher)'

This feature covers also most cases of non-structural dative pronominalisation (see 2.3.), like the "benefactive":

voleva comprarle un anello (a Elena)[+ANIM]	'he wanted to buy **to her** a ring (to Elena)'

the "possessive":

continuano a caderti i capelli (a te)[+ANIM]	'(Your) hair continues to fall out **to you**'

---

[7]Two more categories which are affected by the "clitic climbing", *i. e.* CAUSATIVE constructions such as:

lasciali uscire (i cani) *vs.* *lascia uscirli (i cani)	'let **them** go out (the dogs)'

and PERCEPTIVE constructions such as:

sentendola cantare (Anna) *vs.* *sentendo cantarla (Anna)	'hearing **her** sing (Anna)'

need not be grouped in a closed list because their syntactic behaviour is adequately accounted for by the analysis of their structural properties (see 2.2.).

and the dative selected by polysyllabic prepositions (such as *dietro*, 'behind', *davanti*, 'before', *sopra*, 'above', *sotto*, 'below', *contro*, 'against'):

cercò di correrm**i** *dietro* (*dietro* a me)$_{[+ANIM]}$ 'he tried to run **to me** behind'

The feature SUBCATEGORIZING[pp*a*[DP...]][-ANIM], because of the intuitive closeness between Goal and Dative Case, is responsible for the assignment of the locative pair **{ci, vi}**:

acconsentirv**i** (ad un accordo) $_{[-ANIM]}$ 'to consent **to it** (to an accord)'

abituarc**isi** (all'inquinamento) $_{[-ANIM]}$ 'to get used **to it** (to the pollution)'

costringerv**elo** (alle catene) $_{[-ANIM]}$ 'to constrain **him to it** (to chains)'

The feature SUBCATEGORIZING[pp*(di, da)*[DP...]] corresponds to genitive **{ne}** (see 2.3.):

approfittar**ne** (di lei) 'to abuse **of her**'

dubitar**ne** (della sua sincerità) 'to doubt **of it** (of his sincerity)'

accontentars**ene** (di poco) 'to content himself **of it** (of little)'

accorgers**ene** (del furto) 'to become aware **of it** (of the theft)'

avvisar**ne** i parenti (della scomparsa) 'to inform **of it** the parents (of the loss)'

The feature SUBCATEGORIZING[pp*(in, a, per)*[DP...]][LOCAT] assigns to locative verbs and to verbs of motion the locative list **{ci, vi}**:

essend**oci** già stato (a Parigi) 'having **there** already been (in Paris)'

vado in Italia per viver**ci** 'I am going to Italy to live **there**'

volevo andar**ci** domani (dal dottore) 'I wanted to go **there** tomorrow (to the doctor)'

The scope of this feature is completed by the "SPACE-LOCATIVE" feature, which we shall describe in the next section by addressing a few relevant semantic and pragmatic issues.

2.3. Semantic features

The locative pair **{ci, vi}** can also occur with verbs which are neither locative verbs nor verbs of motion and which therefore do not belong to the aforementioned category. These forms, labelled "SPACE-LOCATIVE" (Calabrese, 1985; pp. 149-153), all express actions or events which presuppose a location in the extra-linguistic reality, because the semantic role of Location is considered to be an essential feature of the actions or events expressed. The feature "SPACE-LOCATIVE" thus completes the feature SUBCATEGORIZING[pp*(in, a, per)*[DP...]][LOCAT], accounting for cases (like *mangiare*, 'to eat' or *dormire*, 'to sleep') in which the link to the locative clitics is semantic rather than structural:

mangiando**ci** tutti i giorni (al ristorante) 'eating **there** every day (at the restaurant)'

puoi dormir**ci** tranquillo (in giardino) 'you can happily sleep **there** (in the garden)'

The second semantic feature identifies the set of verbal forms which carry a component of "MOTION AWAY" in their meaning and thus select a [pp*(da)*[DP...]] expressing "motion from origin, separation". This category is assigned a locative **{ne}**, which we distinguish, following Cordin (1988b), from the partitive and genitive **{ne}**:

vorrei uscir**ne** incolume (dalla vicenda) 'I would like to get out **of it** in one piece'

cadend**one** in malo modo (dal balcone) 'falling badly **from it** (from the balcony)'

allontaniamoc**ene** immediatamente 'let us get away **from it** immediately'

discesone (da cavallo) '(having) dismounted **from it** (from a horse)'

This set of features is completed by two paradigms of idiomatic expressions, in which the clitic appears in a lexicalised form, without accomplishing a real function, as the result of a

phenomenon of "pronominal morphologization" (Berretta, 1985c), in which the clitic loses its pronominal or adverbial value to become a desemanticised morpheme linked to the verb. Considering only the most frequent and non-marked cases, which have been integrated into the system as autonomous lexical entries[8], the model contains:

- PARADIGM (1) → andar**sene**, 'to go away'
- PARADIGM (2) →volerci, 'to be necessary'

This set of morphosyntactic, structural and semantic verbal features provides a satisfactory model of most phenomena of enclitic pronominalization, with the irrelevant exception of the 'dativo etico', a particular kind of non-structural dative which cannot be adequately treated by the feature SUBCATEGORIZING[pp*a*[DP...]][+ANIM] (see 2.2.). The 'dativo etico' (Salvi, 1988) is obligatorily realized by a clitic and indicates the person who is emotionally implied in the event expressed by the verb:

e allora lui, sai cosa fa? **mi** salta giù dalla finestra!

'and you know what he does? he jumps **me** out from the window!'

te lo bevi, un bel caffè?

'why don't you drink **yourself** a nice coffee?'

This construction is usually a proclitic form and thus falls outside the scope of this work. Furthermore, it depends on pragmatic factors, such as the context of the utterance and the expression of the speaker's attitude and emotions. It is indipendent of the morphosyntactic and semantic characteristics of verbs and usually occurs in contexts of informal conversation. It is therefore impossible to treat the 'dativo etico' within our model, which is designed to analyse and produce other kinds of text (more formal and close to written modalities).

3. Conclusions

Once the model of verbal features had been translated into software and implemented, the verbal suffixes of infinitive, gerund, participle and imperative contained in the morphological lexicon ceased to point arbitrarily towards the clitic sub-list and started to operate selectively, considering the morphosyntactic and semantic features of the verbs.

As a result, the analyser deals successfully with most phenomena of enclitic pronominalization. Following the implementation of a morphophonetic and syntactic constraint on the ordering of particles in clitic clusters (Visconti, 1990), the morphological lexicon will furthermore be able to filter agrammatical clitic clusters, such as:

*torglier**lone**	'to take **it from it**'
*avvicinando**sigli**	'getting **himself** nearer **to him**'

allowing acceptable forms only, such as:

dam**melo**	'give **it to me**'
dicendo**glielo**	'telling **it to him**'[9].

---

[8]Such as: andar**sene** = 'to go away'; volerci ≅ 'to be necessary'; entrarci ≅ 'to be relevant'; metterci ≅ 'to take (time)'.

[9]See also: Busch (1985); Evans, Lepschy, Morris et alii (1978); Lo Cascio (1970); Seuren (1974); Wanner (1977).

4. References

Abney S. (1987), *The English Noun Phrase in its Sentencial Aspect*, PhD Dissertation., MIT.

Belletti A. (1993), "Case Checking and Clitic Placement. Three issues on (Italian/Romance) Clitics", in M. Starke (ed.), *Geneva Generative Papers*, University of Geneva, 1, 2; pp. 101-117.

Battaglia S. and Pernicone V. (1954), *La grammatica italiana*, Torino, Loescher; pp. 238-255.

Berretta M. (1985a), "**ci** vs. **gli**: un microsistema in crisi?", Atti del XVII Convegno della SLI (Urbino 1983), *Sintassi e morfologia della lingua italiana d'uso. Teorie e applicazioni descrittive*, Roma, Bulzoni.

Berretta M. (1985b), "Struttura informativa e sintassi dei pronomi atoni: condizioni che favoriscono la 'risalita'", in H. Stammerjohann (ed.), *Tema-Rema in italiano*, Tübingen, Narr; pp. 71-83.

Berretta M. (1985c), "I pronomi clitici nell'italiano parlato", in G. Holtus e E. Radtke (eds.), *Gesprochenes Italienisch in Geschichte und Gegenwart*, Tübingen, Narr; pp. 185-224.

Brunet J. (1978-1985), *Grammaire critique de l'italien*, Paris, Université de Paris VIII-Vincennes, vol. VIII.

Burzio L. (1986), *Italian Sintax. A Government- Binding Approach*, Dordrecht, Reidel.

Busch U. (1985), *Die klitischen Pronomina des Italienischen*, Tübingen, Narr.

Calabrese A. (1985), "La sintassi dei pronomi atoni", in C. Schwarze (ed.), *Bausteine für eine italienische Grammatik*, Band II, Tübingen, Narr; pp. 117-179.

Castelfranchi C. and Parisi D. (1976), "Towards One **si**", *Italian Linguistics*, 2/2; pp. 83- 121.

Chomsky N. (1981), *Lectures on Government and Binding*, Foris, Dordrecht.

Chomsky N. (1992), *A Minimalist Program for Linguistic Theory*, MIT Occasional Papers in Linguistics, 1, MITWPL.

Cinque G. (1976), "Proprio e l'unità del **si**", *Rivista di grammatica generativa*, 1, 2; pp. 101-113.

Cordin P. and Calabrese A. (1988), "I pronomi personali", in L. Renzi (ed.), *Grande Grammatica italiana di consultazione*, Bologna, Il Mulino, 1; pp. 535-592.

Cordin P. (1988a), "I pronomi riflessivi", in L. Renzi (ed.), *Grande Grammatica italiana di consultazione*, cit., pp. 593-603.

Cordin P. (1988b), "Il clitico **ne**", in L. Renzi (ed.), *Grande Grammatica italiana di consultazione*, cit.; pp. 633-641.

Delogu C. (1989), "The Morphological Lexicon of a Speech Recognition System for Italian", *Rivista di Linguistica*, 1, 1; pp. 95-114.

Elia A., Martinelli M. and D'Agostino E. (1981), *Lessico e strutture sintattiche*, Napoli, Liguori.

Evans K.J., Lepschy G.C., Morris S.C. et alii (1978), "Italian Clitic Clusters", *Studi italiani di linguistica teorica ed applicata* 7; pp. 153-168; it. version., "Nessi di clitici italiani", in G.C. Lepschy, *Nuovi saggi di linguistica italiana*, Bologna, Il Mulino, 1989; pp. 85-101.

Laezlinger C. and Wehrli E. (1991), "FIPS: un analyseur interactif pour le français", *T.A. Informations. Revue Internationale du Traitement Automatique des Langues*, 32, 2; pp. 35-49.

Leone A. (1979), "Dal **si** riflessivo al **si** impersonale", *Lingua nostra*, 40; pp. 21-23.

Lepschy A.L. and Lepschy G.C. (1977), *The Italian Language Today*, London, Hutchinson; it. version, *La lingua italiana*, Milano, Bompiani, 1984; pp. 106-112; 181-182; 194-199.

Lepschy G.C. (1974), "Alcune costruzioni con **si**", in *Studi linguistici in onore di Tristano Bolelli*, Pisa, Pacini, pp. 174-184; in *Saggi di linguistica italiana*, Bologna, Il Mulino, 1978; pp. 31-39.

Lepschy G.C. (1976), "Two observations on Castelfranchi and Parisi 'Towards One **si**", *Italian Linguistics*, 2; pp. 157-160.

Lepschy G.C. (1989), "Costruzioni con **si**", in *Nuovi saggi di linguistica italiana*, Bologna, Il Mulino; pp. 103-117.

Lo Cascio V.(1970), *Strutture pronominali e verbali italiane*, Bologna, Zanichelli.

Lo Cascio V. (1974), "Alcune strutture della frase impersonale italiana", in M. Medici e A. Sangregorio (eds.), *Fenomeni morfologici e sintattici nell'italiano contemporaneo*, Atti SLI 7, Roma, Bulzoni, 1; pp. 167-195.

Napoli D.J. (1976), "At least two **si**'s", *Italian Linguistics*, 2/2; pp. 123-148.

Rizzi L. (1976), "Ristrutturazione", *Rivista di grammatica generativa*, 1, 1; pp. 1-54.

Rizzi L. (1993), "Some Notes on Romance Cliticization", ms., University of Geneva.

Salvi G. (1988), "La frase semplice", in L. Renzi (ed.), *Grande Grammatica italiana di consultazione*, cit.; pp. 29-113.

Scalise S. (1983), *Morfologia lessicale*, Padova, CLESP.

Serianni L. (1988), *Grammatica italiana. Italiano comune e lingua letteraria. Suoni forme costrutti*, with the collaboration of A. Castelvecchi, Torino, Utet; pp. 247-261.

Seuren P.A.M. (1974), "Pronomi clitici in italiano", in M. Medici e A. Sangregorio (eds.), *Fenomeni morfologici e sintattici nell'italiano contemporaneo*, Atti SLI 7, Roma, Bulzoni, vol. II; pp. 309-327.

Simone R. (1983), "Punti d'attacco dei clitici in italiano", in F. Albano Leoni et al. (eds.), *Italia linguistica: idee, storia, strutture*, Bologna, Il Mulino; pp. 285-307.

Visconti J. (1990), *Tratti condizionanti un verbo nella selezione delle forme pronominali enclitiche*, Tesi di Laurea, Università di Torino, Torino.

Wanner D. (1977), "On the Order of Clitics in Italian", *Lingua*, 43; pp. 101-128.

Towards an Expert System for Upper Sorbian

EDUARD WERNER

Abstract

Since the use of Sorbian in every-day-life is more or less restricted to the local dialects most Sorbian speakers feel uncertain about orthography and grammar. Existing dictionaries, however, do not help you much to find a word or word form if you do not already have an idea how to write it.

This paper introduces a concept for a computer-based Upper Sorbian dictionary which is capable to handle ill-formed input; it provides the possible correct forms and can cope with compound forms and incongruencies. Furthermore it is the first conception for a monolingual Sorbian dictionary.

1 Introduction

The Sorbs are nowadays the smallest slavic people (about 60,000 speakers, more than half of them are speaking Upper Sorbian). They entirely live in Germany in Sachsen and Brandenburg mainly in the villages around the centers of Cottbus and Bautzen. In Germany, of course, everybody must know the German language while virtually nobody *must* learn Sorbian. Therefore every adult Sorb is able to fluently read, write and speak German whereas the Sorbian language is limited mainly to the speaker's personal environment (home, neighbour, friends). With them the local dialect is spoken. Although Sorbian is also taught at school pupils usually speak German at a higher level than Sorbian, especially when it comes to terminological problems related to sciences, which is mainly due to the fact that sciences are taught in German (also at Sorbian schools) and Sorbian scientific or technical terminologies almost only exist in dictionaries.

There are two Sorbian high schools (one for Upper Sorbian in Bautzen/Budyšin and one for Lower Sorbian in Cottbus/Chośebuz) and no Sorbian university.

This situation leads to several phenomena:

- extended code-switching between German and Sorbian
- interferences between Sorbian and German. Since the Sorbian understand German whilst the German don't understand Sorbian language society takes care of reducing the Sorbian influence on German while strong German influence on Sorbian is more or less accepted, at least in every day talk. As a consequence we find relatively few and locally restricted Sorbian loanwords in German and many German loanwords in Sorbian.
- ad-hoc-loaning from German whenever the Sorbian word is unknown while ad-hoc-loaning from Sorbian into German does not occur. According to what we said above scientific and technological terms are generally unknown, the Sorbian equivalent might even be unintelligible.
- virtually every scientific text is written in German which reduces the use of Sorbian even further

2 About Sorbian Dictionaries

2.1 Existing dictionaries

The Sorbian dictionaries normally available to a Sorbian speaker are Sorbian-German or German-Sorbian dictionaries. All of them are written for German speakers who are learning Sorbian (not vice versa) so it is taken for granted that the Sorbian speaker speaks German sufficiently well. There are no dictionaries at all who use Sorbian as metalanguage and according to this there's nothing that could be compared, say, to the Oxford Advanced Learner's Dictionary. Reducing our point at issue to dictionaries we could even say that Sorbian can't be taught to the Sorbs at the same level as English. Sorbian is taught at Sorbian schools much less intensively than German at German schools or English at English schools. With regard to the fact that many German or English speakers have difficulty correctly spelling words of their own language it can hardly surprise that orthography is a wide-spread problem among the Sorbs.

The only dictionary existing at least partly as a database is the new two-volume German-Sorbian dictionary published in 1989/1991. The database (a dbase file) is conceptually poorly designed (you could say it simply lacks a design) without the possibilities of searching synonyms, antonyms, incorporating pictures or otherwise extending it with no great pain.

2.2 Using a Sorbian dictionary

We suppose you've heard a Sorbian word and want to know how it's written. In this case you must use a Sorbian-German dictionary or a German-Sorbian dictionary. These bilingual dictionaries have several disadvantages:

- You obviously can't look up a Sorbian word in a German–Sorbian dictionary that has no German counterpart (e.g. *ćěpc* 'a part of the traditional catholic women's clothing'). We wouldn't like those words to be left out.
- Using a German–Sorbian dictionary you have to know the German equivalent of the Sorbian word. This is trivial in most, but not in all cases.
- The equivalent given by the dictionary might not be exactly what you wanted due to the fact that there are a lot of almost-exact translations out there but not the real thing. Although you might get away with the translation given by the dictionary it could imply incorrect connotations thus strengthening the German influence on Sorbian.
- Looking up the word in a Sorbian–German dictionary requires to already know how the questionable word is spelled. Especially that you might want to find out by means of a dictionary.

We suppose our typical user to be a person who

- speaks Sorbian sufficiently well (he knows enough words and enough grammar to be able to use a monolingual dictionary)
- knows the meaning of the word he wants to use though he does not necessarily know the German equivalent
- does not know how to spell words right
- is unsure about the paradigm of the word which might be a different one in his local dialect

Of course our dictionary must also be usable by someone who has hit upon an unknown word in a text. But since this is already the goal of the "normal" Sorbian-German dictionaries we'll take it as understood.

3 Sorbian spelling

The Sorbian orthography is somewhat archaic; for a normal speaker who is speaking Sorbian simply because it's his or her mother-tongue Sorbian orthography is almost unpredictable. For those meddling with slavistics it would be predictable (although complicated) if it were consequent.

A short disgression on Sorbian phonology and historical phonology (we shall only deal with Upper Sorbian here) might be helpful: old *g* has become *h* which is still written in most (not in all cases) but often, especially at the end of the word and before consonants, not pronounced; *ł* and *w* are pronounced the same (English *w*) but are of different provenience (*ł* is old hard *l* and *w* is usually old *w*); *ć* and *č* are pronounced the same, but *ć* is old *t* before front vowels while *č* is common slavic *č*. *ch* is pronounced like *k* at the beginning of words and roots. These are the most common problems; there are others related to assimilation, syncope and dialect forms.

To give an example: the Sorbian word for "wasp" is *wosa*. Pronouncing it the same way we might also write *włosa* and get "hair". If we had written *łosa* we'd got the gen. or acc. sg. of "elk", if we had written *hłosa* it would have been the gen. sg. of "voice". The nominatives of these two genitives would be written *łós* "elk" and *hłós* "voice"—and pronounced like *wóz* "car".[1] Of course, those niceties can also occur anywhere else in the word (our example was a really short one) and can turn looking up a word in a dictionary into a nightmare even when you already know which word you are looking for.

4 Finding the correct form of a word

There is another complex of problems due to morphology: from a normal Sorbian word about twenty forms can be derived and a computer-usable dictionary should be able to cope with any of them, since guessing a special form like the infinitive or nominative singular from an unknown word can't be safely done by the user due to possible ambiguities. (Slavic languages are not agglutinative as e. g. Finnish, so there are often alternative interpretations of a word form.)

The main problems with regard to dialect speakers are

1. different paradigms in literary language and vernacular
2. absence of paradigms (e. g. *wotćeć* 'to chop off', *wotetnu, wotetnješ* 1./2. sg. pres)
3. absence of concepts (dual, special forms for male people, aspect)
4. absence of morphonological changes

An example (though a silly one) to illustrate the point: *spěwam spěwaj* can mean "sing to the songs" in which case *spěwam* is dative plural of *spěw* "song" and *spěwaj* imp. sg. (second or third person) of *spěwać* "to sing". It can also mean "I sing two songs"; in this case *spěwam* is 1. sg. pres. act. of *spěwać* and *spěwaj* acc. du. of *spěw*. So you can't tell a priori for either of these forms whether you have to look for a verb or a noun.

What is more, there are significant morphological differences between literary Sorbian and the dialects. In the catholic dialect e. g. *spěwam* can only be dat. pl. of "song" since the 1. sg. of *spěwać* is here *spěwjem*, not *spěwam*. *Ludźi* in the literary language is gen. pl. of *ludźo* "people", in the vernacular it is also the nom. pl.

5 A Concept

We need a dictionary which accepts ill-formed input. As we have seen, many errors can be expected only from speakers of certain dialects so it sounds sensible to ask the user first where he comes from (We might even try to find this out automatically by reading `/etc/passwd`): A speaker of the catholic dialect will easily confuse the writings of *byk* "bull" and *bok* "side" pronouncing both as [bọk], which a speaker

[1] These are only spellings which do have a meaning. We could search for a lot of nonexisting words like *łhosa, whohsa* etc. which would be pronounced the same way if they existed.

of e. g. the eastern territory wouldn't do. So we could have a common database for whole-area-errors and add a this-area-only error database. For the time being we have refrained from that. Since most speakers of the young generation speak the catholic dialect we shall concentrate mainly on the catholic dialect and on typical errors made in school tests.

We shall refrain from checking typos (transposed chars etc.). The actual dictionary must give the right spelling with some additional information such as paradigm or meaning. We'll specify this below. A postprocessor would be useful for pretty-printing or generating special output to be included by other programs.

Now what is our main database expected to do? It should provide

- all possible words which might be meant in their correctly spelled form
- the meanings of the words in order to identify them with cross-references to synonyms and antonyms
- information about the paradigms
- a picture, if sensible
- examples about how and where to use a given form
- idiomatic expressions and proverbs.

The pictures should not only pop up but react to further input as well: looking up the word "head" should could e. g. show a picture of a head where you can click on, say, the ear to receive the appropriate information about this part of the body. This might be relatively easily implemented with XPCE.

The system should be extensible, so you can add new features such as providing the congruent form of an adjective to a given noun without too much pain. Adding sounds may be another desiderate of the future.

To enable other people to write extensions and improvements (and to use it with as little restrictions as possible), the system should be freely available. To ensure the free availability, no commercial software is used.

6 Implementation

The following assumes that the reader is somewhat familiar with Prolog.

6.1 Platform

The hardware is a normal PC (486/66, 16 Meg RAM) running LinuX and X-Windows 11R5. The second test site is a 486/33 with 8 Meg RAM. We have SWI-Prolog with and without XPCE.

6.2 The database

The original idea was to provide only a prolog/XPCE front-end to a "real" database such as OBST or Postgres. Tests with a database[2] containing only 10,000 lexicon entries à 9 stems and 40 endings, however, made both test sites swap (no other users!) so that it took more than a minute to retrieve an entry—without the front-end. So I decided to implement the database myself with regard to the following points:

- minimum main memory requirements (disk space is not a point: a 1 GB hard disk is cheaper than 16 Meg RAM)
- easy implementation and maintenance
- fast retrieval

[2]The database was generated automatically by a perl script and therefore contained only stems like 'aaaa', 'aaab' and so on.

Special multi-user features are not implemented. Since there won't be more than three users writing entries at the same time, the merging capabilities of RCS should be sufficient.

Reducing the amount of memory used is achieved by splitting the database in many parts and to load only the required part. Old clauses are being thrown away so that our process doesn't allocate more and more memory.

Easy implementation and maintenance (including error recovery) means that the data should be ASCII files (prolog clauses). If the disk space matters, you can pipe them through `gzip`; this will take only slightly longer because the files have to be relatively small to meet the first request.

The increase of speed was overwhelming: the retrieval time sank to about 0.5 sec. What's more, the retrieval time was invariant of the size of the database: retrieving word forms from a test database with 250,000 lexicon entries à 11 stems and 40 endings was not slower than retrieving from a small database.

The data files live in three directories: `KÓNCOWKI`, `ZDONKI` and `LEKSIKON`. (These are the sorbian words for *endings*, *stems* and *lexicon*.) The `KÓNCOWKI` directory contains files with prolog clauses that contain endings of a certain class. The file is named after the class. To give an example: the endings of a conjugation class called `nje` will be found in `KÓNCOWKI/nje.pl` and the data look like:

```
ending(nje, nogen, 'jem.', 'u', '1.', nom, sg, ag, pres, aa,werb).
```

This defines a fact `ending/11` which contains information about the name of the paradigm (`nje`), an additional piece of information which says that this clause is not to be used when generating forms (`nogen`), the input ending (`jem.`), the output ending (`u`), the person (`1.`), the case (`nom` for nominative; to enable an implementation of a congruency check between subject and verb, finite verb forms are regarded as nominatives), the number (`sg` for 'singular'), the gender (`ag` stands for 'any gender'), the tense or mode[3] (`pres` stands for present tense), the aspect[4] (`aa` for 'any aspect'), and the word class (`werb` for 'verb').

Many endings are coded twice or more (when they can belong to different paradigms, tenses, persons). Coding each ending for each paradigm differently makes the database easier to maintain since it avoids side effects and requires only little additional memory on disk.

Since there are more lexicon entries and stems than endings the directories `ZDONKI` and `LEKSIKON` contain subdirectories `a/`, `b/` `c/` and so on which in turn contain subsubdirectories `aa/`, `ab/` etc. So you find all the stems starting with *pad* in `ZDONKI/p/pa/pad.pl`. If the stem is shorter than three letters it will be closer to the root; a stem *z* will be found in `ZDONKI/z/z.pl`. (Note that we haven't yet defined *stem*. For the purpose of our parser a stem is simply the front part of a word. If the word starts with the regular negation prefix *nje-* the stem is the front part of the word after the *nje-*.)

The use of stems is restricted; you cannot take any stem to generate any form of a word. Therefore the stems contain information about person, case, aspect, gender and so on. In `ZDONKI/p/pa/pad.pl` you will find clauses like:

```
word(werb, ag, p, ap, ac, anr, pres, ang, nc, noaf,'', '.pan','padn', nje, 'padnyć').
```

The information contained in this fact are the word class (`werb`), the gender (`ag` = 'any gender'), the aspect (`p` = perfective), the case (`ac` = 'any case'; `nom` would not be so good since we might want to generate participles which can also have other cases), the number (`anr` = 'any number'), the tense or mode (`pres` = 'present tense'), the negation (`anr` = 'any negation' means negative and positive forms can be derived from this stem), the comparability (`nc` = 'not comparable': we do not favour comparatives derived from verbs or participles), the possibility of building an analytic future (`noaf` = 'no analytic future'), a list of obligatory morphemes[5], the input stem (here: `'.pan'`), the output stem (here: `padn`, the paradigm, and the related lexicon entry (here: `'padnyć'`). You will have remarked that the input stem and input ending have a leading resp. trailing dot the output stem and output ending have not. This dot is used for special patterns and will be explained below (see 8).

This is only one of the stems referring to the lexicon entry `'padnyć'`. On the whole there are, due to morphological and morphonological changes and restriction of usage, eleven. By coding every stem

[3] There's no use to distinguish here since you can't have, say, an imperative perfect or konjunctive future in Sorbian.

[4] This is something slavic; verbs can be perfective or imperfective which means more or less the act expressed is regarded as a whole or as something in progess. The point is simply that you cannot build any form from any aspect.

[5] The purpose is to be able to correct user input when parsing e. g. reflexivatantum: **prašem** should not be corrected to **prašam**, but to **so prašam**.

once and not encoding morphonological and morphological changes our database is very modular and little error-prone due to side-effects. Trying to code the whole verb in one stem would have resulted in something like `paj1d2(n1y)1` in which `j1` would have been an epenthetical *j* occuring in the imperative before stem-final *n*, `d2` would have been a *d* that can be dropped before *n* and interchange with *dź* before *e*, `n1` is *n* that becomes palatalized under certain conditions and `(...)1` means that this complex can be dropped in the l-forms but not in the last syllable. The merit to have all the information together is doubtful since it couldn't be done for the really irregular words anyway, and the implementation wouldn't be so easy any more.

The entries for gender, tense etc. in `ending/11` and `word/15` needn't be identical, but they must be compatible. Our example facts (they are compatible) would allow `'.panjem.'` to be interpreted as a first person singular presence perfective; the concatenation of output stem and output ending is *padnu*, the correct written form which means 'I fall'. The lexicon entry belonging to this word is `padnyć`.

The corresponding lexicon entries can be found in `LEKSIKON/p/pa/pad.pl` (I think the system is clear now). This file contains all the lexicon entries starting with *pad* in prolog clauses like:

```
entry('padnyć',werb, 'padnyć p:  here comes the explanation text', [List_of_stems],
    [List_of_sources], [List_of_idioms], [List_of_proverbs], [List_of_synonyms],
                    [List_of_antonyms],[List_of_pictures]).
```

`List_of_stems` needn't be a complete list of all stems, but it must contain enough stems to generate every correct form. This information is needed to generate alternate forms to a given form that might be irregular. `List_of_pictures` isn't used yet, it will contain a list of graphics related to the given lexicon entry.

7 The user interface

The proper user interface that makes use of graphics, hypertext features and so on is yet to be written. At the present you simply get a prompt to enter a word form. The input string is converted into a list of atoms (the words you typed in). This list is given to the main retrieval routine which returns a list of interpretations. The interpretations are verified and pretty-printed and the user will be prompted for the next input.

8 The preprocessor and the pattern processor

The goal of the preprocessor and the pattern processor is to find certain patterns of letters in the words and replace them by others. The difference between the two is that the preprocessor performs those transformations that *always* occur while the pattern processor performs transformations that *can* but needn't occur. The coding of both is identical except that the preprocessor has an additional final cut to prevent backtracking.

First of all, to the word a leading and a trailing dot are added. They are markers of the beginning and the end of the word and needed by patterns. After that several letters will be replaced, thus reducing the number of patterns used by the pattern processor:

ł	→	w (same pronouncation)
č	→	ć (same pronouncation)
x	→	ks (*x* does not exist in Sorbian)
ch	→	x (patterns with *c* and *ch* differ
.ch	→	.k
ź	→	ž (*ź* exists only after *d*)
ń	→	jn (pronouncation in relevant positions)

Finally, duplicates are removed and the result is given to the pattern processor.

While the preprocessor gives exactly one solution to every input the pattern processor can give more than one since it performs backtracking. The output of the pattern processor will be caught in a list with the "corrected" possible misspellings. Here we take care of possible changes like $dn \rightarrow n$, $h \rightarrow \emptyset$ and others. A simple example: Suppose the user entered `paahnjem` the preprocessor would pass `.pahnjem.`

to the pattern processor which would return `.panjem.` and `.pahnjem.`. These two words we would try to analyze, for the first one we would succeed.

9 How data is retrieved from the database

Retrieving the data is straightforward. According to the first letters of the processed word the name of the stem file is determined which is loaded if it is not by chance already in memory. If no stem matches (`concat(Stem,_,Processed_word)` fails), the search fails. If a stem matches (which means that `concat(Stem,_,Processed_word)` succeeds), the endings belonging to this stem are loaded. If no compatible ending is found, so that `concat(Stem,Ending,Processed_word)` succeeds, the search proceeds with the next stem. If the search is successful, the found information (lexicon entry, correct stem and ending, person, case, number, gender, mode, aspect etc.) is added to the list of solutions. This list is passed to the output routine which pretty-prints the correctly spelled words, generates additional forms and the additional information.

10 A sample session

?- parser.

Zapodaj słowo abo 'quit' za kónc posedźenja:

¿ sóm njeprašaw

so prašeć ip: sej wot někoho wotmołwu, informaciju žadać

Idiomatiske wurazy:
so prašeć kaž (jako) připołdnica: so wjele a dołho prašeć

njejsym so prašał
1. sg perfekt mask ip
1. sg perfekt žiw ip
1. sg perfekt wos ip

Zapodaj słowo abo 'quit' za kónc posedźenja:

¿ budžech panoł

padnyć p: wot grawitacije ćěrjeny so zemi bližić, zwjetša spěšnje a tohodla, zo sy runowahu abo zepěru zhubił; z padoruneje do wodoruneje pozicije přińć

Přisłowa:
na jedyn raz njepadnje wjaz: to so njehodźi spěšnje wotbyć
na jedyn rub njepadnje dub: (to samsne)
na jedyn rub hola (štom) njepadnje: (to samsne)

padnyć p: wumrěć (wo wojakach we wójnje)

budźech padnył
1. sg konjunktiw mask p
1. sg konjunktiw žiw p
1. sg konjunktiw wos p
bych był padnył

Zapodaj słowo abo 'quit' za kónc posedźenja:

¿ quit

Yes
?-

11 Things to do

- Complete implementation of stems and endings and lexicon entries
- Tools for database maintenance and consistency check. (You perhaps can't expect a normal linguist to write prolog clauses.)
- A better user interface, allowing for hypertext features, text attributes, implementation of graphics. This will most probably done with XPCE.

List of Participants

JÓZSEF ANDOR
Department of English, Janus Pannonius University
Ifjúság u. 6.
Pécs, HUNGARY, H-7624
andor@btk.jpte.hu

JÖRG ASMUSSEN
The Danish Dictionary, University of Copenhagen
Njalsgade 80
Copenhagen 5, DENMARK, DK-2300
Fax: 45 3154 2595

ILONA BELLOS-MORISSON
Larousse PLC
43-45 Annadale Street
Edingburgh, SCOTLAND U.K., EH7 4AZ

VLADIMIR BENKO
Computational Linguistics Laboratory, Faculty of Education
Comenius University
Moskovska 3
Bratislava, SLOVAKIA, 81334

ELISABETH BLANCHON
Centre de Terminologie et de Néologie INALF-CNRS-URA IS 76
Université Paris Nord, Ave J. B. Clément
Villetaneuse, FRANCE, 93430

CHRISTOPH BLASI
Bibliographisches Institut F.A. Brockhaus AG (Duden Publishers)
Postfach 100311
Mannheim, GERMANY, D-68167
Phone: 0049621 3901336

STEPHAN BOPP
Free University of Amsterdam SR Lexicologie
De Boelelaan 1105
Amsterdam, THE NETHERLANDS, 1081 HV
lexico@let.vu.nl
Phone: xx31/20/548 37 31
Fax: xx31/20/661 30 54

LORNE H. BOUCHARD
Départment de Mathématiques et d'Informatique
B.P.8888, Succursale Centre-Ville
Montréal (Québec), CANADA, H3C 3P8
lhb@info.uqam.ca

ANNA BRAASCH
Centre for Spragteknologi, Kobenhavos Universitet
Njalsgade 80
Copenhagen S, DENMARK, DK-2300

JEREMY BUTTERFIELD
Harper Collins Publishers, Bilingual Dictionaries
P. O. Box 4 OND
Glasgow, SCOTLAND U.K.
jeremy@collins.co.uk

M. TERESA CABRÉ
Universitat MPEU Fabra
Rambla Sta Nica, 30.
Barcelona, SPAIN, 08002
cabre@upf.es

OLIVER CHRIST
Universität Stuttgart, Institut für Maschinelle Sprachverarbeitung
Azenbergstr. 12
Stuttgart 1, GERMANY, D 70174
oli@ims.uni-stuttgart.de

JEREMY CLEAR
COBUILD Ltd
Westmere, 50 Edgbaston Park Road
Birmingham, UNITED KINGDOM, B15 2RX

KINGA CSENGERY
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250

MARC DOMENIG
Institut für Informatik der Universität Basel
Petersgraben, 51
Basel, SWITZERLAND, CH-4051

MARKUS DUDA
Research Group for Computational Linguistics
Humboldt University Berlin
Unter den Linden 6
Berlin, GERMANY, D-10 099
duda@compling.hu-berlin.de

ANNIBALE ELIA
Via Bernini 88
Napoli, ITALY, 80129

LOUISETTE EMIRKANIAN
Département de linguistique
B.P. 8888, Succursale Centre-Ville
Montréal (Québec), CANADA, H3C 3P8
lhb@info.uqam.ca

STEFANO FEDERICI
ILC-CNR (Pisa)
Via Della Faggiola 32.
Pisa, ITALY, 56100

DAVID GAATONE
French Department Tel-Aviv University
Tel-Aviv, ISRAEL, 69978
Fax: 3-6409457

GUNTER GEBHARDI
Lehrstuhl für Computerlinguistik, Fachbereich Germanistik
Humboldt Universität zu Berlin
Unter den Linden 6
Berlin, GERMANY, D-10 099
gebhardi@compling.hu-berlin.de

GREGORY GREFENSTETTE
Rank Xerox Research Centre, Grenoble Laboratory
Meylan, FRANCE, 38240
grefen@xerox.fr

MAURICE GROSS
LADL, Université Paris 7
2 place Jussieu, Cedex 05
Paris, FRANCE, 75221
mgross@ladl.jussieu.fr

CLAUDE GRUAZ
CNRS Paris
5 rue aux Boulangers
Avrilly, FRANCE, 27240
Phone: (16) 32 57 41 88
Fax: (16) 32 67 43 53

PATRICK HANKS
Oxford University Press
Walton Street
Oxford, UNITED KINGDOM, OX2 6DP
phanks@oup.co.uk

ULRICH HEID
Universität Stuttgart, Institut für Maschinelle Sprachverarbeitung
Azenberg Straße 12
Stuttgart, GERMANY, D-70174
uli@ims.uni-stuttgart.de

VÁCLAVA HOLUBOVÁ
Institute for Czeh Language AV CR
Letenska 4
Praha 1, CZECH REPUBLIC, 118 51
klimova@site.cas.ez

MATTHIAS KAMMERER
Hirschstraße 119
Karlsruhe, GERMANY, D-76137
Phone: 0721/817409

PATRICIA KELLY
OURIA, Royal Irish Academy
19 Dawson Street
Dubun 2, IRELAND
curiapk@ccvax.vcd.ie

FERENC KIEFER
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250

ADAM KILGARIFF
Longman Dictionaries
Longman House Burnt Mill
Harlow, Essex, UNITED KINGDOM, CM20 2JE
100347.76@compuserve.com

GÁBOR KISS
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250

LAJOS KISS
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250

JANA KLIMOVÁ
Institute for Czech Language, Czech Academy of Sciences
Letenska' 4
Praha 1, CZECH REPUBLIC, 118 51
klimova@cspguk11.bitnet

FRANK KNOWLES
Institute for the Study of Language and Society, Aston University
Birmingham, UNITED KINGDOM, B4 7ET
f.e.knowles@aston.ac.uk

HEINZ-DETLEV KOCH
Department of Computational Linguistics, University of Heidelberg
Karlstraße 2
Heidelberg, GERMANY, D-69171
Phone: 00496221 543248
Fax: 00496221 543242

TRUUS KRUYT
Institute for Dutch Lexicology INL
P. O. Box 9515
Leiden, THE NETHERLANDS, 2300 RA
kruyt@vulcri.leidenuniv.nl

BEATRICE LAMIROY
Department of Linguistics K. Universiteit Leuven
Blyde Inkjomstraat 21
Leuven, BELGIUM, 3000
bl%users%lw@cc3.kuleuven.ac.be

ERIC LAPORTE
CERIL, Institut Gaspard-Monge, Université de Marne la Vallée
2 rue dela Butte-Verte
Noisy-le-Grand, FRANCE, F-93166
eric@pixel.univ-mlv.fr

ANDREA LEHR
Germanistisches Seminar, University of Heidelberg
Hauptstraße 207-209
Heidelberg, GERMANY, D-69117
Phone: 00496221 543253
Fax: 00496221 543257

YVETTE MATHIEU
LADL, Université Paris 7
2 place Jussieu, Cedex 05
Paris, FRANCE, 75221
mathieu@ladl.jussieu.fr

MEHRYAR MOHRI
LADL-IGM Université Marne la Vallee
Paris, FRANCE
mohri@univ-mlv.fr

NAM JEE-SUN
LADL Université Paris 7
2, Place Jussieu, Cedex 05
Paris, FRANCE, 75221

OLE NORLING-CHRISTENSEN
The Danish Dictionary, University of Copenhagen
Njalsgade 80
Copenhagen 5, DENMARK, DK-2300
Fax: +45 3154 2595

JUDIT PAIS
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250

JÚLIA PAJZS
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250
pajzs@nytud.hu

MIREILLE PIOT
LADL, Université Paris 8
30, Rue Chapon
Paris, FRANCE, 7500
mpiot@ladl.jussieu.fr

VITO PIRELLI
ILC-CNR (Pisa)
Via della Faggiola 32
Pisa, ITALY, 56126
perimila@vm.cnuce.cnr.it

GÁBOR PRÓSZÉKY
MORPHOLOGIC
Fő u. 56-58. I/3.
Budapest, HUNGARY, H-1011

ROSWITHA RAAB-FISCHER
Institut für Englische Sprache und Literatur, Englisches Seminar I.
Albert-Ludwigs-Universität
K. G IV Rempart Str. 15.
Freiburg i.Br., GERMANY, D-79085

PETER ROE
Institute for the Study of Language and Society, Aston University
Birmingham, UNITED KINGDOM, B4 7ET
p.j.roe@aston.ac.uk

FERENC ROVNY
Kossuth L. University Foreign Language Centre Computerized
Lexical-Terminological Database Centre
P. O. Box 41
Debrecen, HUNGARY, H-4010
rovny@tigris.klte.hu

IRENE ŠĚRAK
Sorbisches Institut e. V.
Bahnhofstraße 6
Bautzen, GERMANY, D-02625
eduard@kaihh.hanse.de

PASI TAPANAINEN
Rank Xerox Research Centre, Grenoble Laboratory
Meylan, FRANCE, 38240
tapanai@xerox.fr

LÁSZLÓ TIHANYI
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250

TSCHEKE TIBOR
Stürtz Electronic Publishing GmbH (STEP)
Technologiepark Kettelerstraße
Rimpar, GERMANY, D-97222
step@step.de

BALDASARE TUZZO
Faeolta' di Magistero, Universita' Degli Studi di Palermo
Viale Michelangelo, 1752
Palermo, ITALY, 90145
tuzzo@cvc.unipa.it

SIMEONETTA VIETRI
Instituto di Linguistica
Fisciano
Salerno, ITALY, 84100

ILDIKÓ VILLÓ
Research Institute for Linguistics Hungarian Academy of Sciences
P. O. Box 19
Budapest, HUNGARY, H-1250

JACQULINE VISCONTI
Dép. des Langues et des Littératures Romanes
Faculté des Lettres - Université de Genéve
1211
Genéve, SWITZERLAND, 4-CH
visconti.uniza.unige.ch

DUSKO VITAS
Faculty of Mathematics
Studentski Trg 16
Belgrade, YUGOSLAVIA (SERBIA), 11000

ELISABETH VITIELLY
Sanofi Recherche, Centre de Montpellier
371, Rue du Professeur Blayac, Cedex 04
Montpellier, FRANCE, 34184

EDUARD WERNER
Sorbisches Institut. V.
Bahnhofstr. 6
Bautzen, GERMANY, D-02625
eduard@kaihh.hanse.de